



Rzut pionowy – ćwiczenie w Excelu

Wstęp

Wiemy, że każde ciało, które spada swobodnie z wysokości h , porusza się z przyspieszeniem równym przyspieszeniu ziemskiemu ($a = g$). Przyspieszenie ziemskie jest skierowane pionowo w dół, do środka Ziemi. Jego wartość jest różna w różnych miejscach naszej planety ze względu na niejednorodności w budowie skorupy ziemskiej. Do celów obliczeniowych przyjmuje się najczęściej wartość $g = 9,81\text{m/s}^2$.

Powiedzieliśmy sobie, że wszystkie ciała spadają z jednakowym **przyspieszeniem**, ale jednak z codziennych obserwacji wynika, że nie wszystkie spadają z jednakową **prędkością**. Na przykład, człowiek ze spadochronem spada wolniej niż bez spadochronu, piórko spada wolniej niż kamień (łatwo można to sprawdzić upuszczając oba przedmioty jednocześnie z tej samej wysokości). Dlaczego tak się dzieje, skoro Ziemia przyciąga wszystkie przedmioty jednakowo? Odpowiedzią jest **siła oporu powietrza**, która przeciwdziała sile grawitacji. My w naszych rozważaniach dotyczących ruchu ciał w powietrzu przyjmiemy obraz nieco uproszczony, gdyż **nie będziemy uwzględniać sił oporu powietrza**. W rzeczywistości odpowiadałoby to sytuacji, w której rozważalibyśmy ruch ciał znajdujących się w doskonałej próżni poruszających się w polu grawitacyjnym naszej planety.

Opis ruchu ciała w rzucie pionowym

Jeżeli nadamy jakiemuś ciału prędkość początkową \vec{v}_0 w kierunku pionowym w górę, to w całym czasie ruchu ciało ma stałe przyspieszenie ziemskie \vec{g} zwrócone pionowo w dół, a więc w stronę przeciwną do prędkości początkowej \vec{v}_0 . Zatem ciało, wznosząc się, wytraca stałe prędkość, aż do chwilowego zatrzymania się – wtedy prędkość chwilowa $v = 0$. W tym momencie ciało osiąga maksymalną wysokość H . Ta pierwsza faza ruchu w górę jest ruchem jednostajnie opóźnionym ze stałym opóźnieniem równym g . W drugiej fazie ruchu ciało swobodnie spada.

Położenie ciała w kierunku pionowym (współrzędna y), czyli jego wysokość w dowolnej chwili t wyrażamy za pomocą wzoru analogicznego do wyrażenia na drogę w ruchu jednostajnie opóźnionym ($s = s_0 + v_0t - \frac{at^2}{2}$):

$$y(t) = y_0 + v_0t - \frac{gt^2}{2} \quad (1)$$

Najwyższą wysokość, na którą wzniesie się ciało, H_{max} , czyli **zasięg pionowy**, ciało uzyska po czasie wznoszenia t_w danym wzorem:

$$t_w = \frac{v_0y}{g} \quad (2)$$

H_{max} możemy otrzymać ze wzoru (1) wstawiając t_w w miejsce t . Wykonując przekształcenia otrzymujemy kolejno:

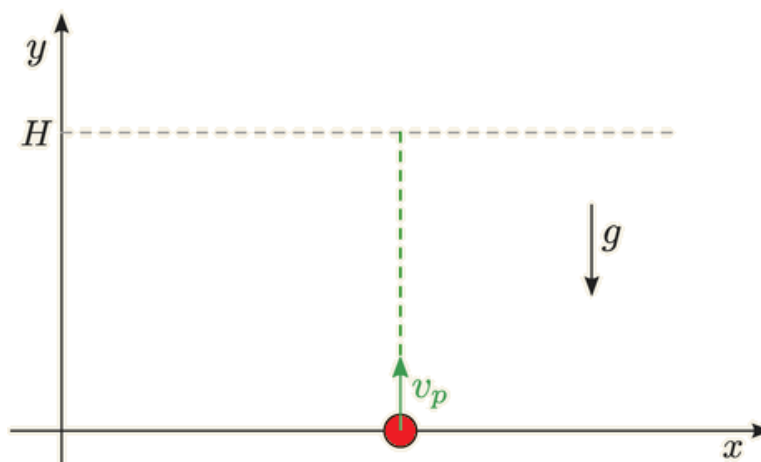
$$H_{max} = y_0 + v_0t_w - \frac{gt_w^2}{2} = y_0 + v_0 \frac{v_0y}{g} - \frac{g \left(\frac{v_0y}{g}\right)^2}{2} = y_0 + \frac{v_0^2}{g} - \frac{v_0^2}{2g}$$

i ostatecznie

$$H_{max} = y_0 + \frac{v_0^2}{2g} \quad (3)$$

Zależność (3) opisuje maksymalną wysokość ciała rzuconego w górę z prędkością v_0 .





Rysunek 1. Rzut pionowy w górę.

W przypadku, kiedy ciało rzucamy z takiej samej wysokości, na jaką ono spadnie, $t_w = t_s = \frac{1}{2}t_c$, gdzie t_s – czas spadania, t_c – czas całkowity ruchu. Kiedy wyrzut następuje z innej wysokości, niż ta, na której ciało spadnie, wówczas $t_c = t_w + t_s$ ale $t_w \neq t_s$.

W przypadku ogólnym, czas spadania t_s , wyznaczymy ze wzoru:

$$H_{max} = \frac{gt_s^2}{2}$$

stąd otrzymujemy

$$t_s = \sqrt{\frac{2H_{max}}{g}} \quad (4)$$

Zależność prędkości v od t – czasu ruchu dana jest zależnością:

$$v(t) = v_0 - gt \quad (5)$$

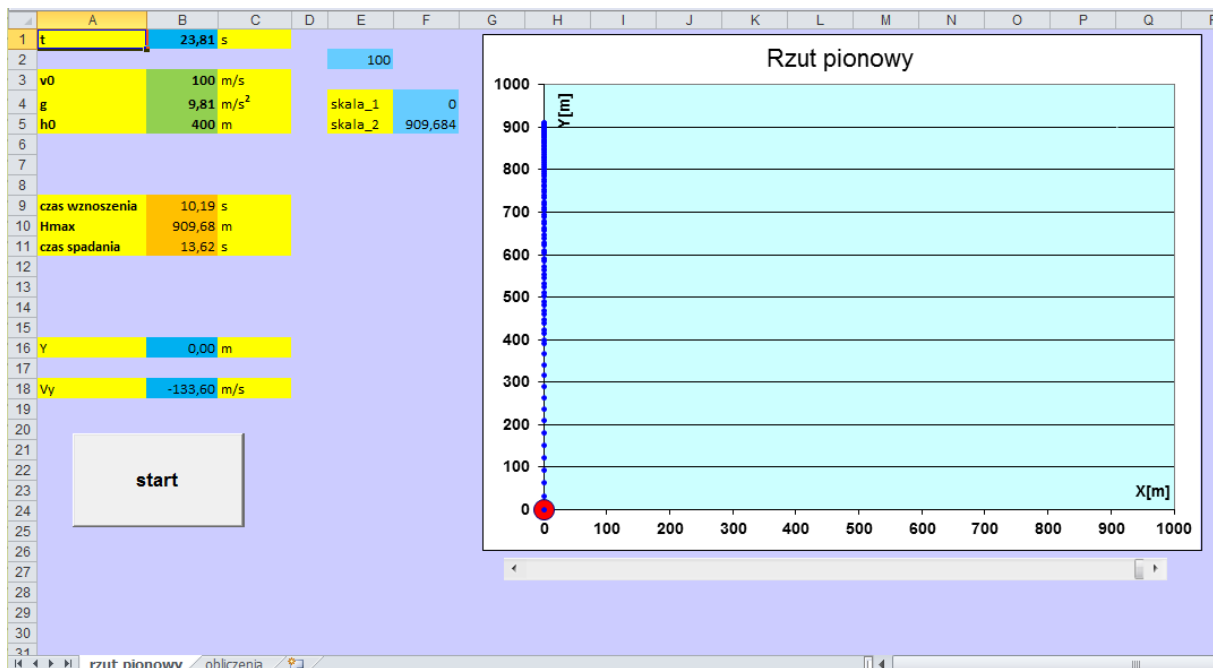
Łatwo możemy się przekonać, że **rzut pionowy** jest szczególnym przypadkiem **rzutu ukośnego**, w którym ciało rzucone jest pod kątem $\alpha = 90^\circ$ do poziomu.

Jak działa model rzutu pionowego w Excelu

W arkuszu *rzut pionowy* umieszczony został wykres przedstawiający tor rzutu pionowego w układzie współrzędnych (X, Y) (rys. 2). Czas lotu ciała został podzielony na 100 równych przedziałów. Wykres w postaci małych granatowych kropek przedstawia kolejne położenia ciała – od momentu wyrzucenia do jego upadku (wszystkie potencjalne położenia), natomiast duża czerwona kropka pokazuje bieżące położenie ciała (dla czasu t widocznego w komórce B1).

W komórkach kolumny B sformatowanych na zielono znajdziesz wejściowe parametry rzutu do sterowania wykresem, i tak: w komórce B3 – prędkość początkową v_0 wyrażoną w m/s, B5 – wysokość początkową h_0 (wartość równa zero odpowiada poziomowi ziemi). W komórce B4 umieściliśmy wartość przyspieszenia ziemskiego.

Dla wprowadzonych przez użytkownika parametrów do komórek B3:B5, w komórkach B9:B11 (za pomocą formuł, które opiszemy niżej) obliczane są stałe wartości charakteryzujące ruch, takie jak czas wznoszenia i spadania oraz maksymalna wysokość. W komórkach B16 i B18 znajdują się wartości chwilowe, takie jak: bieżące położenie (współrzędna y) oraz prędkość w kierunku pionowym v_y .



Rysunek 2. Rzut pionowy – model w Excelu

Wprowadź przykładowe wartości do komórek B3:B5 (mogą być takie, jak na rysunku 2), a następnie naciśnij przycisk **start**. Przyciskiem tym uruchomisz *makro*, które przygotuje i przeprowadzi symulację rzutu pionowego, wyznaczając bieżące wartości położenia podczas wznoszenia się i spadania – y – tabela wartości znajduje się w arkuszu *obliczenia*. Wartości z arkusza *obliczenia* zostaną wyświetlone na wykresie ukazując przebieg ruchu, czyli tor poruszającego się ciała.

Makro – jest to program napisany w języku VBA (Visual Basic for Application), który wykonuje zapisane w nim instrukcje automatyzując nam wykonywanie pewnych czynności. Kod makra możemy tworzyć i modyfikować w specjalnie do tego celu przeznaczonym edytorze stowarzyszonym z produktami MS Office. Więcej informacji o makrach znajdziesz w *Dodatku* umieszczonym na końcu instrukcji.

Całkowity czas ruchu (czyli suma czasu wznoszenia i czasu spadania: $t_c = t_w + t_s$) zostanie w tej symulacji podzielony na 100 równych przedziałów – w komórce E2 możesz obserwować aktualną wartość kroku, czyli bieżący przedział trwania symulacji. Bieżąca wartość czasu symulacji, wyrażona w sekundach, wyświetlana jest w komórce B1.

Poniżej wykresu umieściliśmy pasek przewijania, który jest powiązany z komórką E2 i pozwala na zmianę wartości z zakresu od 0 do 100 – odpowiadają one poszczególnym przedziałom trwania symulacji. Zmieniając wartość przedziału za pomocą paska zaobserwuj jak zmienia się czas lotu ciała w komórce B1.

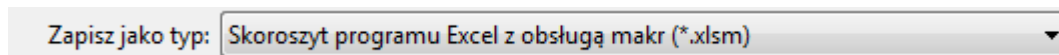
Zwróć uwagę, że jeśli będziesz klikać na strzałkach paska przewijania, na wykresie będzie się przemieszczała duża czerwona kropka ukazując bieżące położenie ciała wyrzuconego z parametrami podanymi w komórkach B3:B5.

Sprawdź, że zmiana jakiegokolwiek wartości w komórkach B3:B5 (sformatowanych na zielono) spowoduje wyzerowanie wartości w komórce E2 oraz tabeli z wartościami x i y w arkuszu *obliczenia*, co będzie skutkowało również „wyzerowaniem” wykresu – nie będzie widoczny wykres składający się z małych granatowych kropek.

Jeśli chcesz z powrotem zobaczyć na wykresie pełny tor, czyli przeprowadzić pełną symulację rzutu pionowego z zadanymi parametrami, kliknij ponownie na przycisku **start**.

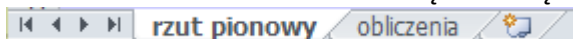
Wykonanie

Do wykonania symulacji rzutu pionowego potrzebny będzie pusty plik Excela. Zapisz go na dysku swojego komputera pod nazwą *rzut_pionowy_symulacja*. Poniżej pola z nazwą pliku znajdziesz pole *Zapisz jako typ* (Rysunek 3). Wybierz w nim *Skoroszyt programu Excel z obsługą makr*. Wykonanie tej akcji spowoduje, że plik otrzyma rozszerzenie *.xslm* zamiast typowego rozszerzenia *.xlsx*.



Rysunek 3. W polu *Zapisz jako typ* wybierz „Skoroszyt programu Excel z obsługą makr”

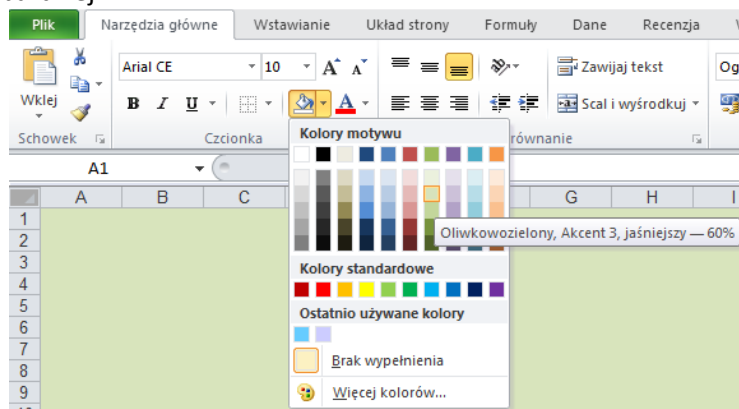
W pliku *rzut_pionowy_symulacja.xslm* zmień nazwę pierwszego arkusza na *rzut pionowy*, drugiemu nadaj nazwę *obliczenia*. Pozostałe arkusze możesz usunąć – nie będą nam potrzebne.



Rysunek 4. Nazwy arkuszy w pliku *rzut_pionowy_symulacja*

Tworzenie symulacji rzutu pionowego rozpoczniemy od przygotowania danych w arkuszu *rzut_pionowy*. Wykonaj w nim następujące czynności:

1. Zaznacz cały arkusz (na przykład używając kombinacji klawiszy Ctrl+A) i za pomocą ikony *Kolor wypełnienia* znajdującego się na wstążce na karcie *Narzędzia główne*, w grupie *Czcionka*, zmień kolor tła komórek na dowolny, ale jasny!, odcień koloru niebieskiego, fioletowego lub zielonego (Rysunek 5) – wybierz ten, który lubisz najbardziej.



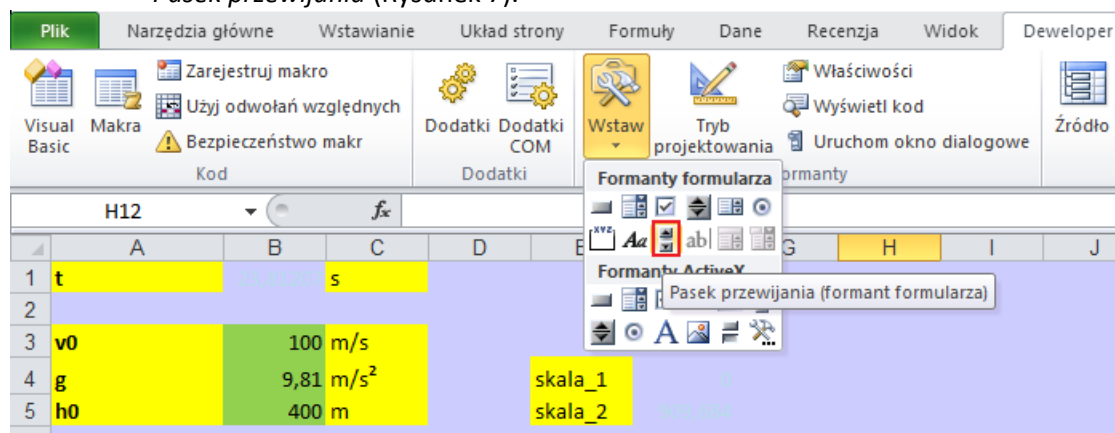
Rysunek 5. Zmiana tła komórek w arkuszu *rzut_pionowy*

2. Spójrz na rysunek 2 zaprezentowany powyżej. Komórki z żółtym tłem na rysunku 2 nie zawierają żadnych obliczeń. Znajdują się w nich wyłącznie etykiety tekstowe, czyli opisy różnych parametrów niezbędnych do wykonania symulacji. Zajmiemy się teraz ich wprowadzeniem. W tym celu postępuj następująco: do komórki A1 wpisz tekst „t”, do komórki A3 – „v0” itd. W ten sposób wpisz do arkusza wszystkie etykiety. Tło komórek zawierających etykiety sformatuj na żółto (rys. 6).

	A	B	C	D	E
1	t	22,8120	s		
2					100
3	v0	10	m/s		
4	g	9,8	m/s ²		skala_1
5	h0	40	m		skala_2
6					
7					
8					
9	czas wznoszenia	10,1326	s		
10	Hmax	305,68	m		
11	czas spadania	12,8120	s		
12					
13					
14					
15					
16	y		m		
17					
18	Vy	13,28	m/s		

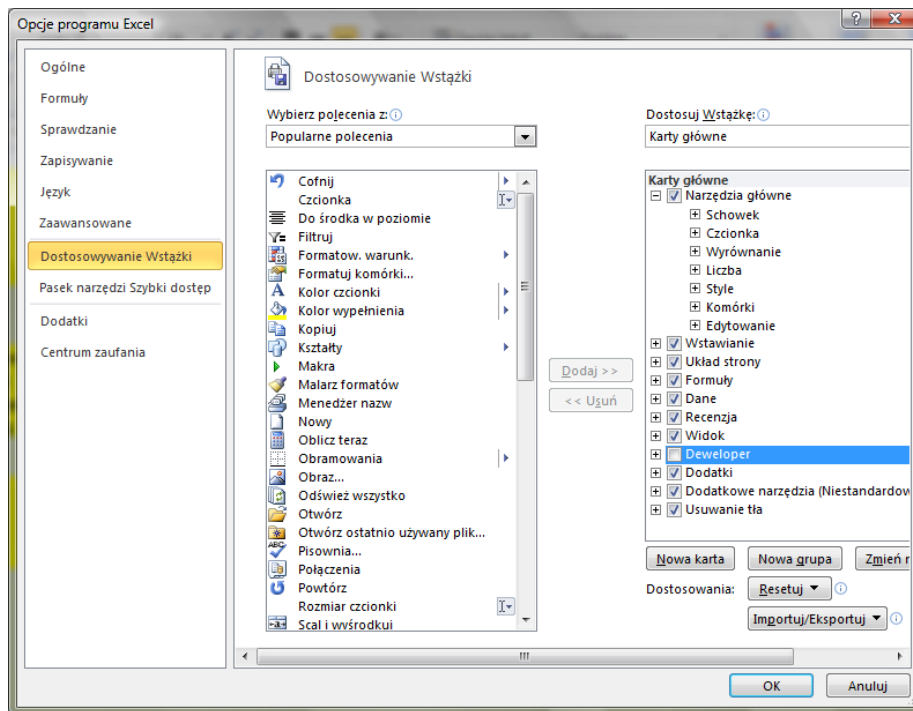
Rysunek 6. Wygląd arkusza rzut_pionowy po wykonaniu pkt. 1 i 2 instrukcji

- Teraz wprowadzimy parametry, które, jako użytkownik, będziesz mógł modyfikować: do komórki B3 wprowadź wartość 100, B4: 9,81 i do B5 wartość 400. Tło komórek sformatuj na zielono.
- Spójrz jeszcze raz na rysunek 2. Poniżej wykresu znajduje się pasek przewijania, postaraj się wstawić go mniej-więcej tym samym miejscu, co w pliku wzorcowym (np. w wierszu 27, w zakresie kolumn od G do Q). W tym celu na karcie *Deweloper* (jeśli nie widzisz jej na swojej wstążce, w następnym punkcie opisałismy jak ją wyświetlić), w grupie *Formanty* znajdź polecenie *Wstaw* i z wewnętrznej listy wybierz formant *Pasek przewijania* (Rysunek 7).



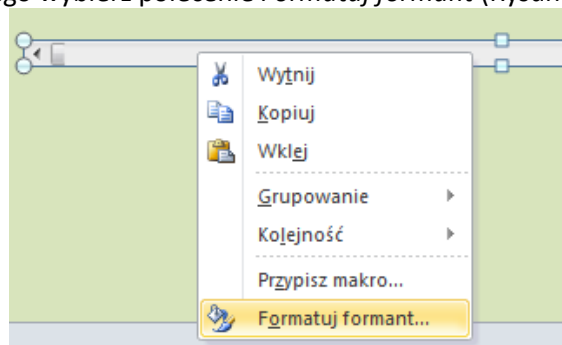
Rysunek 7. Wstawianie paska przewijania

- Uwaga:** jeżeli na wstążce nie ma karty *Deweloper*, włącz ją w następujący sposób: wybierz polecenie *Plik*, a następnie *Opcje*. W oknie *Opcje programu Excel* kliknij *Dostosowywanie wstążki*. Po prawej stronie okna znajdziesz spis wszystkich kart wstążki. Włącz wyświetlanie karty *Deweloper* (Rysunek 8). Od tej pory będzie ona stała widoczna na Twojej wstążce.



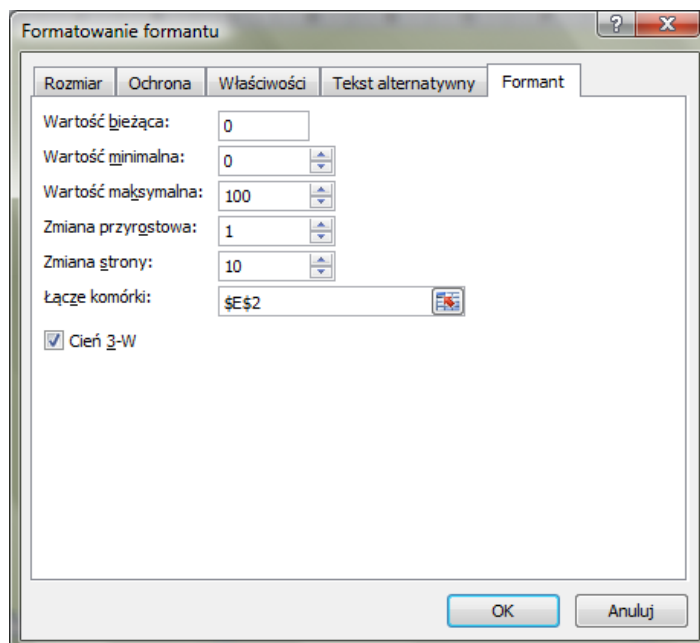
Rysunek 8. Włączanie karty Deweloper

6. „Narysuj” pasek przewijania w odpowiednim miejscu arkusza (spójrz na rysunek 2). Następnie kliknij prawym przyciskiem myszki na pasku przewijania i z menu podręcznego wybierz polecenie *Formatuj formant* (Rysunek 9).



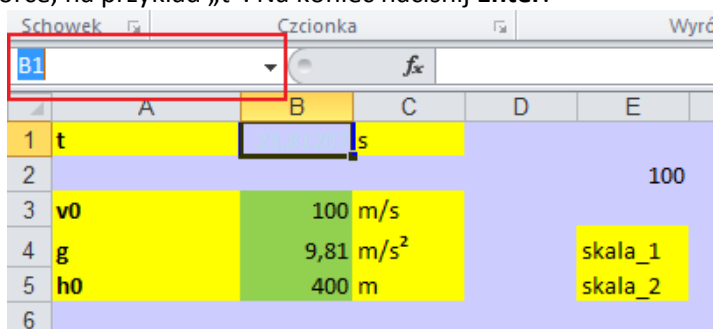
Rysunek 9. Z menu podręcznego wybierz polecenie *Formatuj formant*

7. W oknie *Formatowanie formantu* wprowadź parametry jak na rysunku 10. Pasek będzie obsługiwał wartości od zera (*Wartość minimalna*) do 100 (*Wartość maksymalna*). W komórce E2 (jej adres wprowadź w polu *Łącze komórki*) będzie się wyświetlała bieżąca wartość paska przewijania, czyli jedno ze stu położenia ciała w naszej symulacji rzutu pionowego. Podczas klikania na strzałki paska przewijania, wartości w komórce E2 będą się zmieniały o 1 (świadczy o tym ustawienie w polu *Zmiana przyrostowa*). Po kliknięciu „w pole” paska (czyli gdziekolwiek pomiędzy strzałkami), wartość w komórce E2 zmieni się o 10 (pole *Zmiana strony*). Zatwierdź wprowadzone ustawienia przyciskiem **OK**.



Rysunek 10. Parametry paska przewijania

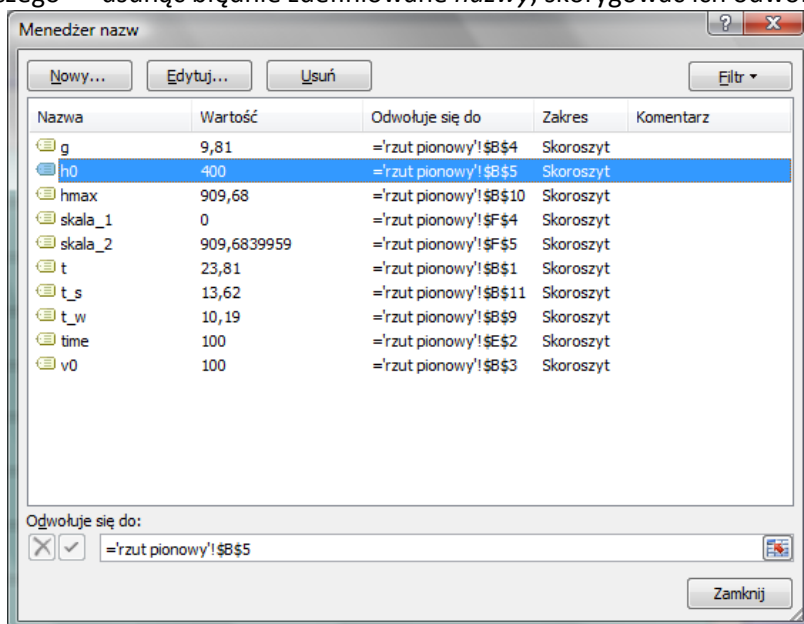
8. Odkliknij zaznaczenie paska (kliknij w dowolnym miejscu arkusza, aby pozbyć się zaznaczenia z paska przewijania). Wypróbuj jego działanie. Zaobserwuj, jak zmieniają się wartości w komórce E2.
9. Kolejnym krokiem tworzenia symulacji będzie wprowadzenie formuł do arkusza. Najpierw jednak, dla wygody, niektórym komórkom nadamy *nazwy*.
Uwaga: Odwołanie do komórki za pomocą nadanej jej własnoręcznie *nazwy* jest alternatywą do odwołania się do niej za pomocą adresu. Będziesz mieć wybór – możesz w formułach odwoływać się do komórki za pomocą jej adresu lub za pomocą *nazwy*.
10. Komórce B1 nadamy nazwę „t”. Uaktywnij komórkę B1 (kliknij na niej). Zwróć uwagę, że adres aktywnej komórki pojawił się w tzw. *Polu nazwy* (pole otoczone czerwoną obwódką na rysunku 11). Kliknij w *Polu nazwy* i wpisz nazwę, którą chcesz nadać komórce, na przykład „t”. Na koniec naciśnij **Enter**.



Rysunek 11. Lokalizacja *Pola nazwy*, za pomocą którego nadajemy *nazwy* komórkom

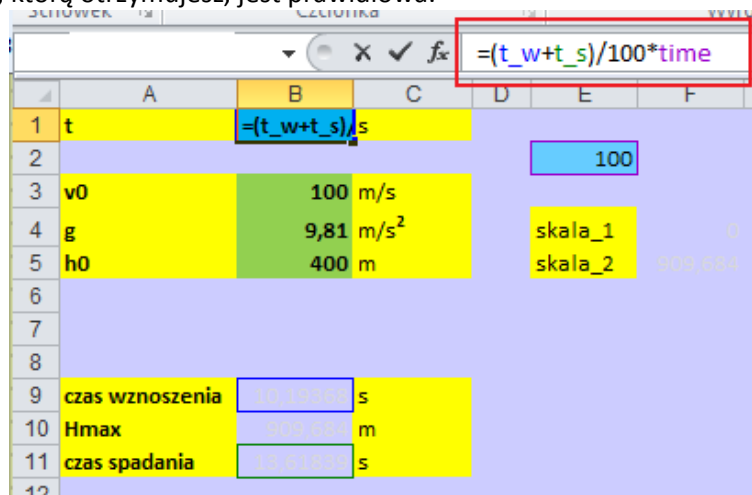
11. Zaznacz komórkę B3 i postępując w sposób opisany w poprzednim punkcie nadaj jej nazwę *v0*.
Uwaga: jeśli zechcesz nadać komórce nazwę wielocłonową, nie używaj spacji! Zamiast niej możesz użyć znaku podkreślenia, np. *kąt_alfa*, lub wpisać nazwę w taki sposób: *KątAlfa* – bez odstępów, ale zachowując czytelność nazwy.
12. W podobny sposób nadaj nazwy pozostałym komórkom, i tak, komórkę B4 nazwij „g”, B5 – „h0”, B9 – „t_w”, B10 – „hmax”, B11 – „t_s”, E2 – „time”, F4 – „skala_1”, F5 – „skala_2”.
Uwaga: jeśli pomylił się definiując *nazwy*, na wstążce na karcie *Formuły* w grupie

Nazwy zdefiniowane znajdziesz ikonę *Menedżer nazw*. Otwiera ona okno z listą wszystkich nazw zdefiniowanych w bieżącym skoroszytcie. Możesz w nim – „w razie czego” – usunąć błędnie zdefiniowane nazwy, skorygować ich odwołanie itp.



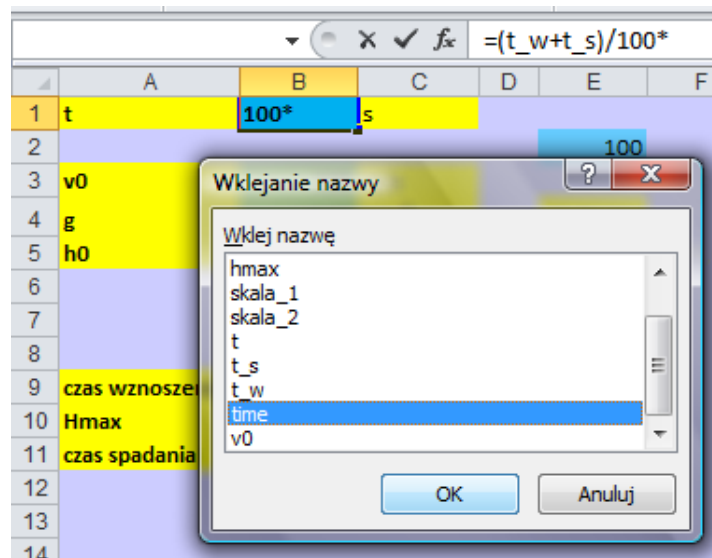
Rysunek 12. Okno *Menedżera nazw* z listą zdefiniowanych nazw, ich bieżące wartości (u Ciebie na tym etapie wykonywania ćwiczenia większość wartości będzie równa zero), odwołania do komórek itp.

13. Najwyższa pora zająć się obliczeniami! Na początek w komórce B1 wyznaczmy czas (wyrażony w sekundach), jaki upłynął od momentu wyrzucenia ciała. Wprowadź do komórki B1 formułę, jak na rysunku 14. Spójrz na swój *Pasek formuły* i sprawdź czy formuła, którą otrzymujesz, jest prawidłowa.



Rysunek 13. Formuła w komórce B1

14. **Uwaga:** tworząc formułę, możesz adresy komórek (lub nazwy) wpisywać ręcznie lub klikać na odpowiednich komórkach – wówczas ich adresy (nazwy) w formule pojawią się automatycznie. W przypadku nazw możesz alternatywnie skorzystać z okna *Wstawianie nazwy*, które zawiera spis wszystkich nazw występujących w skoroszytcie. Okno *Wstawianie nazwy* wywołasz naciskając klawisz funkcyjny **F3**.



Rysunek 14. Tworzenie formuły z użyciem okna *Wstawianie nazwy*

15. W komórce B9 obliczymy czas wznoszenia ciała rzuconego w górę wg wzoru (2).

$$f_x = v_0/g$$

Rysunek 15. Formuła w komórce B9

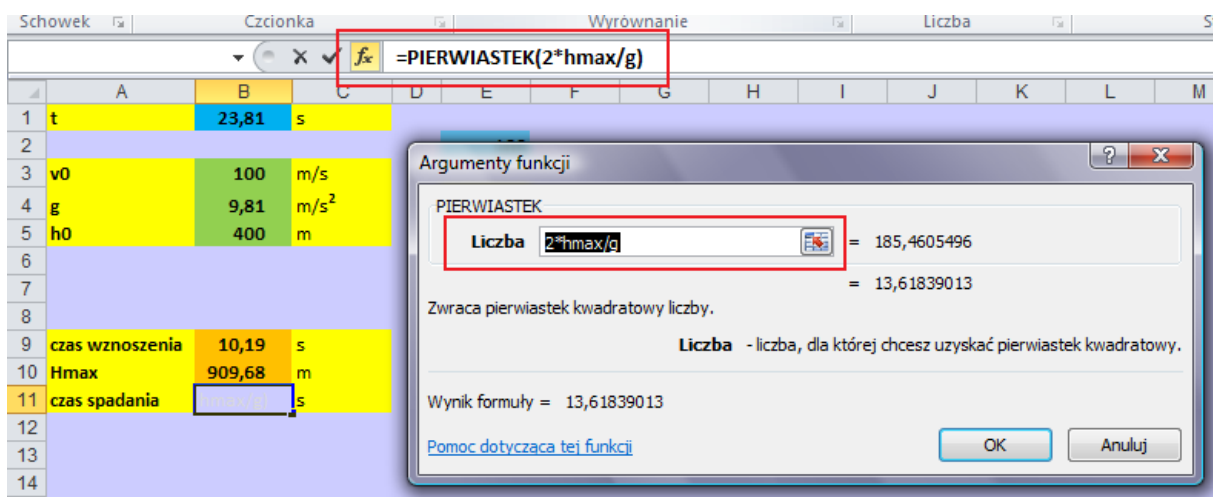
16. Do komórki B10 wprowadź formułę do obliczenia maksymalnej wysokości (3), jaką osiągnie ciało w rzucie pionowym:

$$f_x = h_0 + (v_0^2)/(2*g)$$

Rysunek 16. Formuła w komórce B10

17. W komórce B11 obliczymy czas spadania t_s z wysokości H_{max} według wzoru (4). We wzorze tym występuje pierwiastek. Aby obliczyć pierwiastek w Excelu musimy zastosować funkcję PIERWIASTEK (z kategorii *Matematyczne i trygonometryczne*). Możemy wpisać ją ręcznie, albo wstawić ją korzystając z narzędzi do wstawiania funkcji. Przybliżymy tutaj ten drugi sposób. Aby przygotować formułę z rysunku 17, zaznacz komórkę B11, a następnie kliknij na ikonie *Wstaw funkcję* f_x , którą znajdziesz z lewej strony *Paska formuły*. W oknie *Wstawianie funkcji* wybierz kategorię *Matematyczne*. Otrzymasz spis funkcji należących do tej kategorii. Znajdź i wybierz funkcję PIERWIASTEK(), a następnie w oknie funkcji wprowadź odpowiednie argumenty. Gotową formułę zatwierdź naciskając **OK**.

Uwaga: zwróć uwagę, że kiedy wysokość początkowa h_0 w komórce B5 jest równa zero (nie ma przesunięcia wysokości, ciało spada na takiej samej wysokości z jakiej jest wyrzucane), wówczas $t_w = t_s$ – czas wznoszenie jest równy czasowi spadania.



Rysunek 17. Formuła w komórce B11

18. Współrzędna y opisująca chwilowe położenie ciała w komórce B16:

$$f_x = h_0 + v_0 * t - g * (t^2) / 2$$

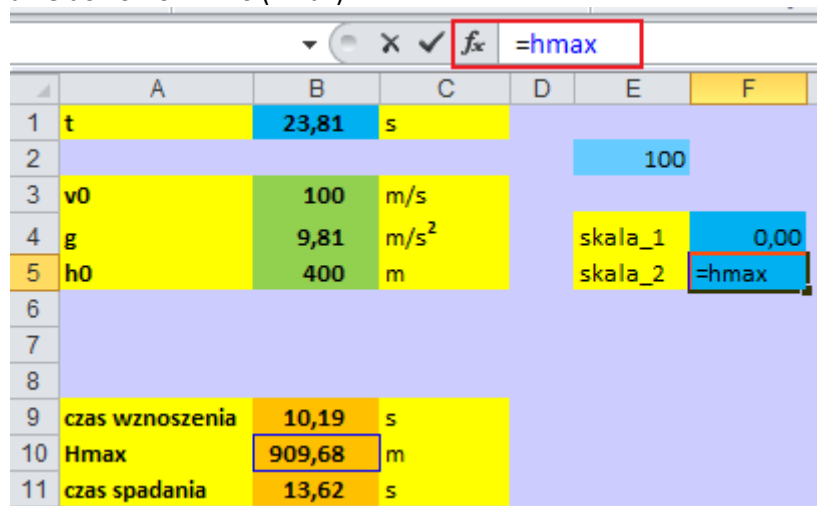
Rysunek 18. Formuła w komórce B16

19. Bieżąca wartość prędkości – v_y (komórka B18):

$$f_x = v_0 - g * t$$

Rysunek 19. Formuła w komórce B18

20. W komórkach F4 i F5 znajdują się wartości, które umożliwią nam odpowiednio wyskalować wykres. Do komórki F4 wpisz 0 (zero). W komórce F5 wprowadź odwołanie do komórki B10 (hmax).



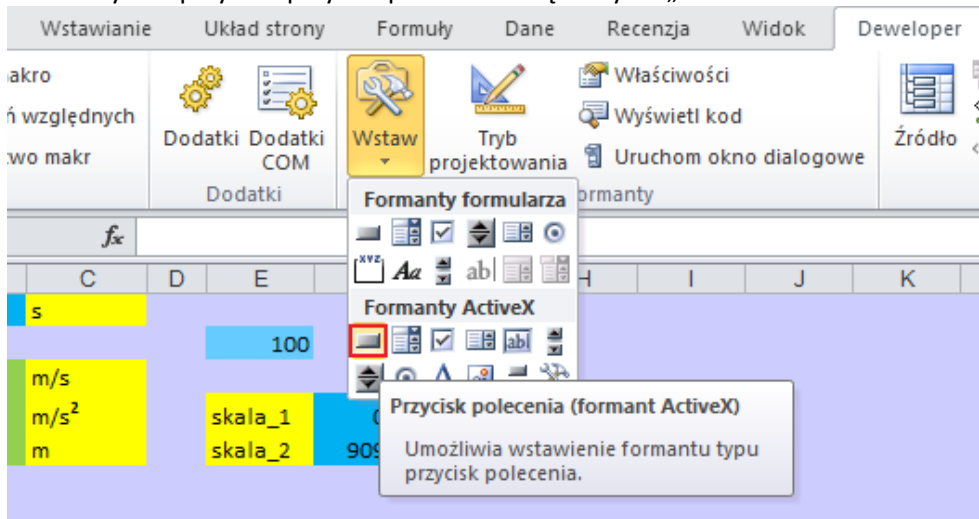
Rysunek 20. Formuła w komórce F5

21. Do tej pory zajmowaliśmy się wyłącznie arkuszem *rzut pionowy*. Teraz przygotujemy dane w arkuszu *obliczenia*, w zakresie komórek A1:B101, które będą bazą do wykonania wykresu obrazującego ruch ciała w rzucie pionowym. Dane w arkuszu *obliczenia* będą wyliczane za pomocą makra (dla parametrów wprowadzonych przez użytkownika w arkuszu *rzut pionowy*).

Przygotowania rozpoczniemy od wstawienia przycisku w arkuszu *rzut pionowy*, który będzie uruchamiał makro. Aby wstawić przycisk przejdź na kartę *Deweloper*, w sekcji *Formanty* kliknij na ikonie *Wstaw* i z wewnętrznej listy, z grupy *Formanty ActiveX*, wybierz *Przycisk polecenia* (Rysunek 21). „Narysuj” przycisk poniżej komórek

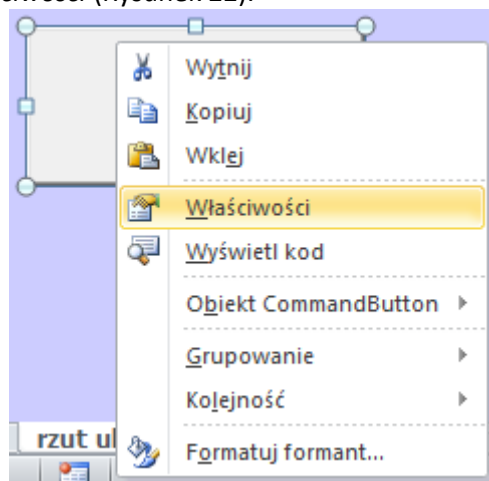
z danymi, gdzieś w zakresie wierszy 22-25 i kolumn A-C. Zwróć uwagę czy na karcie *Developer* jest włączony przycisk *Tryb projektowania*. Jeśli nie – włącz go!

Uwaga: Kiedy przycisk *Tryb projektowania* na karcie *Developer* jest włączony możesz bez trudu modyfikować właściwości formantów, kiedy jest wyłączony – formanty, czyli na przykład przycisk polecenia – są w trybie „działania”.



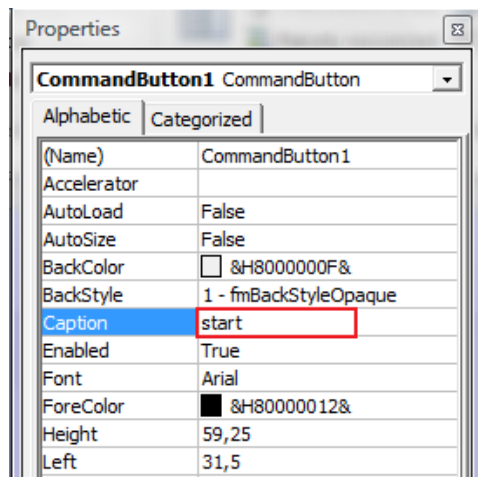
Rysunek 21. Wstawianie Przycisku polecenia, który będzie uruchamiał makro

22. Kliknij prawym klawiszem myszki na wstawionym przycisku i z menu podręcznego wybierz *Właściwości* (Rysunek 22).



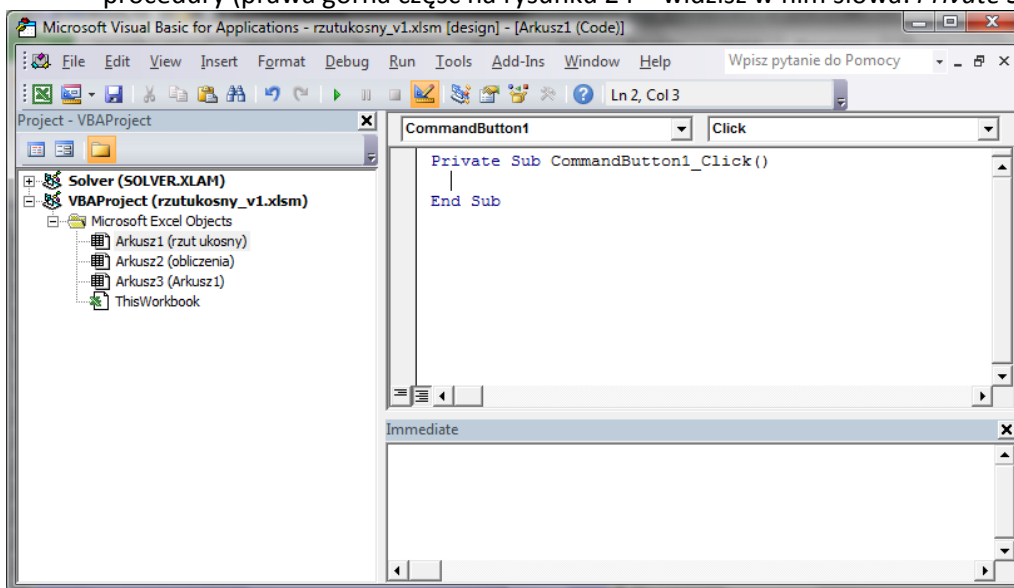
Rysunek 22. Menu podręczne Przycisku polecenia

23. W oknie *Properties*, którego fragment widzisz na rysunku 23, w sekcji *Caption* wpisz słowo „start”. Zamknij okno właściwości (nie musisz zatwierdzać wprowadzonych zmian). Wykonanie tych czynności spowoduje, że na przycisku pojawi się słowo „start”.



Rysunek 23. Właściwości Przycisku polecenia

24. Upewnij się, czy przycisk *Tryb projektowania* na karcie *Deweloper* jest włączony. Jeśli nie jest – włącz go! Kliknij teraz szybko dwa razy na przycisku **start**. Powinien pojawić się edytor kodu VBA (Rysunek 24). Okno edytora domyślnie składa się z kilku części, z których – z naszego punktu widzenia – najistotniejsze są: okno projektu widoczne po lewej stronie (*Project – VBA Project*) oraz okno służące do wpisywania kodu procedury (prawa górna część na rysunku 24 – widzisz w nim słowa: *Private Sub* itd.).



Rysunek 24. Okno edytora VBA

25. Aby dokładnie zrozumieć kod makr, które zaprezentujemy, i dowiedzieć się „o co w ogóle chodzi z tym VBA” powinieneś teraz zapoznać się ze specjalnym *Dodatkiem* zamieszczonym na końcu niniejszej instrukcji.
26. W części służącej do programowania, w miejscu, gdzie widoczny jest kursor, wpisz kod przedstawiony na rysunku 25. Jest to procedura, która uruchomi się po kliknięciu na przycisku **start** umieszczonym w pierwszym arkuszu.

```
Worksheet Change
Private Sub CommandButton1_Click()
    For i = 0 To 100
        ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = 0
        ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = 0
    Next i

    For i = 0 To 100
        ThisWorkbook.Worksheets(1).Cells(2, 5) = i
        Range("A1").Activate
        ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = 0
        ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = ThisWorkbook.Worksheets(1).Cells(16, 2)
    Next i
End Sub
```

Rysunek 25. Kod procedury, która wyliczy współrzędne x, y położenia ciała w rzucie pionowym

W procedurze z rysunku 25 widzisz dwie pętle *For ... Next*. Pierwsza z nich wykona się 101 razy (zmienna *i* – licznik pętli – będzie przyjmowała wartości od zera do 100, w każdym kroku licznik będzie zwiększał się o 1). W pętli zostaną wykonane następujące czynności: zostaną wyzerowane komórki w kolumnach A i B, wierszach od 1 do 101, w drugim arkuszu bieżącego skoroszytu.

Druga pętla *For ... Next* wykona się taką samą ilość razy, jak pierwsza. Pierwsza instrukcja w tej pętli spowoduje, że do komórki o numerze wiersza 2 i numerze kolumny 5 (chodzi oczywiście o komórkę E2) pierwszego arkusza (czyli *rzut pionowy*) będzie podstawiana bieżąca wartość licznika. Zwróć uwagę, że w tej komórce znajduje się informacja o kolejnym numerze przedziału trwania symulacji (cała symulacja została podzielona na 100 przedziałów czasowych). Druga instrukcja oznacza po prostu: „zaznacz komórkę A1 w aktywnym arkuszu”. Instrukcja trzecia to podstawianie kolejno do komórek A1:B101 wartości 0 w miejsce współrzędnych *x*. Czwarta instrukcja podstawia współrzędne *y* położenia ciała w rzucie pionowym wyliczane w komórkach B16 w arkuszu pierwszym.

27. Po przepisaniu procedury z rysunku 25 ustaw kursor tekstowy poza procedurą (poniżej słów kluczowych *End Sub*) i za pomocą list wyboru znajdujących się ponad edytorem tekstu wybierz pozycję *Worksheet* z listy po lewej stronie i *Change* – po prawej stronie. W wyniku wykonania tych czynności na dole okna powinna pojawić się procedura *Worksheet_Change*. Zadaniem tej procedury, która uruchomi się automatycznie po zmianie wartości w którejkolwiek z komórek sformatowanych na zielono w pierwszym arkuszu, będzie wyzerowanie wartości komórek w zakresie A1:B101 w arkuszu drugim.

Uwaga: jeżeli przy okazji, niechcący, w edytorze pojawi się jakaś inna procedura, zaznacz ją (tak, jak w dowolnym edytorze tekstu) i skasuj klawiszem **Delete**.



The screenshot shows the VBA editor with the 'Worksheet' dropdown menu selected and the 'Change' event chosen. The code in the main window is as follows:

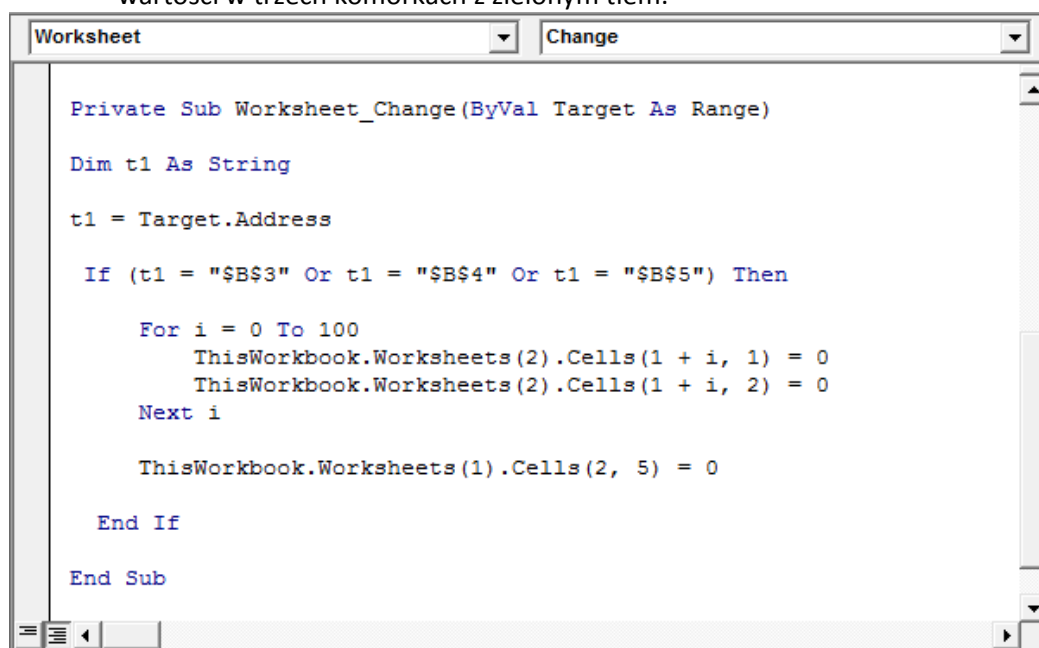
```
Private Sub CommandButton1_Click()  
    For i = 0 To 100  
        ThisWorkbook.Worksheets(2).cells(1 + i, 1) = 0  
        ThisWorkbook.Worksheets(2).cells(1 + i, 2) = 0  
    Next i  
  
    For i = 0 To 100  
        ThisWorkbook.Worksheets(1).cells(2, 5) = i  
        Range("A1").Activate  
        ThisWorkbook.Worksheets(2).cells(1 + i, 1) = 0  
        ThisWorkbook.Worksheets(2).cells(1 + i, 2) = ThisWorkbook.Worksheets(1).cells(16, 2)  
    Next i  
End Sub
```

Below this code, a red box highlights the following procedure:

```
Private Sub Worksheet_Change(ByVal Target As Range)  
End Sub
```

Rysunek 26. Wybór procedury zdarzeniowej Worksheet_Change

28. Pomiędzy słowami kluczowymi (*Sub* i *End Sub*) procedury *Worksheet_Change* wprowadź kod zamieszczony na rysunku 27. Poprzednia procedura nie zawierała żadnych argumentów, tutaj występuje zmienna *Target*, która jest typu zakres (*Range*). Zaprogramujemy procedurę tak, aby „reagowała” wyłącznie na zmianę wartości w trzech komórkach z zielonym tłem.




The screenshot shows the VBA editor with the 'Worksheet' dropdown menu selected and the 'Change' event chosen. The code in the main window is as follows:

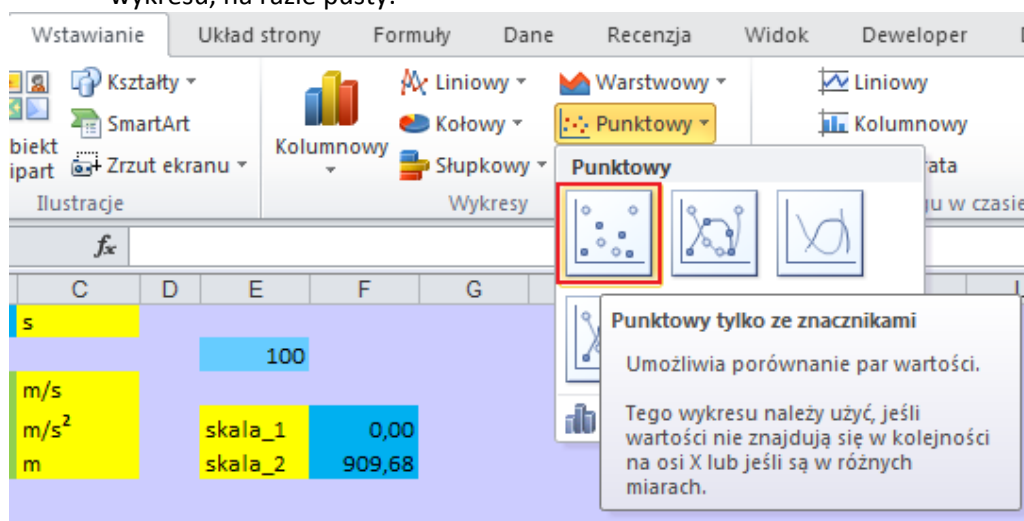
```
Private Sub Worksheet_Change(ByVal Target As Range)  
  
    Dim t1 As String  
  
    t1 = Target.Address  
  
    If (t1 = "$B$3" Or t1 = "$B$4" Or t1 = "$B$5") Then  
  
        For i = 0 To 100  
            ThisWorkbook.Worksheets(2).Cells(1 + i, 1) = 0  
            ThisWorkbook.Worksheets(2).Cells(1 + i, 2) = 0  
        Next i  
  
        ThisWorkbook.Worksheets(1).Cells(2, 5) = 0  
  
    End If  
End Sub
```

Rysunek 27. Kod procedury Worksheet_Change

29. W pierwszej linii kodu została zadeklarowana zmienna *t1* jako zmienna typu tekstowego (*String*). Instrukcja *t1 = Target.Address* oznacza: „zmienniej *t1* przypisz adres komórki (zakresu komórek), w której nastąpiła zmiana wartości”. Instrukcja warunkowa *If ... End If* sprawdza warunek, czyli, czy zmiana nastąpiła w jednej z komórek z zakresu z zielonym tłem (chodzi o komórki B3 lub B4 lub B5). Jeśli warunek jest spełniony, pętla *For ... Next* (znanej nam doskonale z poprzedniej procedury) zeruje wartości w tabelce w arkuszu drugim.

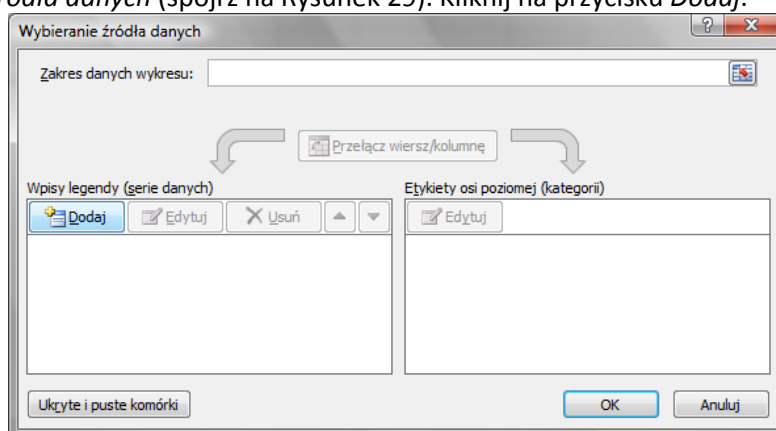
Instrukcja `ThisWorkbook.Worksheets(1).cells(2, 5) = 0` wstawia dodatkowo wartość 0 do komórki E2.

30. Po wpisaniu kodu obu procedur, zapisz wszystko – możesz w tym celu użyć standardowej ikony  dostępnej na pasku narzędziowym lub kombinacji klawiszy **Ctrl+S**.
31. **Uwaga:** Zwróć uwagę, że edytor VBA jest odrębnym oknem. Możesz je zamknąć w standardowy sposób. Aby ponownie wywołać okno edytora VBA naciśnij kombinację klawiszy **Alt+F11** (**F11** to klawisz funkcyjny, znajdziesz go w górnej części klawiatury). W każdej chwili możesz z edytora VBA przejść do skoroszytu Excela (bez zamykania go) używając *Paska zadań* (na dole monitora).
32. Mając wykonane wszystkie obliczenia zajmiemy się przygotowaniem wykresu. Zaznacz dowolną pustą komórkę arkusza *rzut pionowy* znajdująca się poza obszarem danych (na przykład H2).
33. Przejdź na kartę *Wstawianie*. W grupie *Wykresy* wybierz typ *Punktowy*, a następnie podtyp *Tylko ze znacznikami* (Rysunek 28). W arkuszu powinien pojawić się obszar wykresu, na razie pusty.



Rysunek 28. Wybór wykresu, typ Punktowy

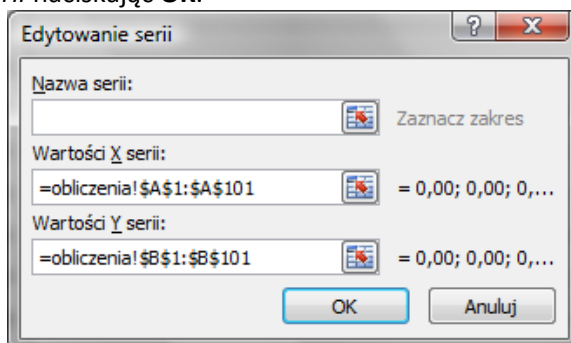
34. Zwróć uwagę czy obszar wykresu jest aktywny. Jeśli tak, przejdź na kartę *Projektowanie* (znajdącą się na końcu wstążki, w grupie *Narzędzia wykresów*) i z grupy poleceń *Dane* wybierz *Zaznacz dane*. Na ekranie pojawi się okno *Wybieranie źródła danych* (spójrz na Rysunek 29). Kliknij na przycisku *Dodaj*.



Rysunek 29. Dodawanie serii danych do wykresu

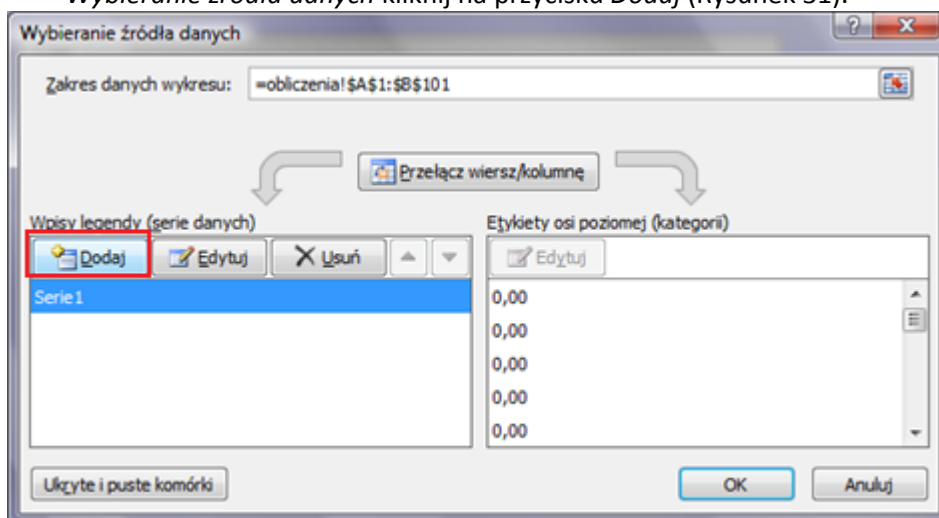
35. W oknie *Edytowanie serii* (Rysunek 30) pole *Nazwa serii* pozostaw puste. Ustaw kursor w polu *Wartości X serii*, przejdź na arkusz *obliczenia* i zaznacz zakres komórek

od A1 do A101. Następnie przestaw kursor do pola *Wartości Y serii*, przejdź na arkusz *obliczenia* i zaznacz zakres komórek od B1 do B101. Potwierdź ustawienia w oknie *Edytowanie serii* naciskając **OK**.



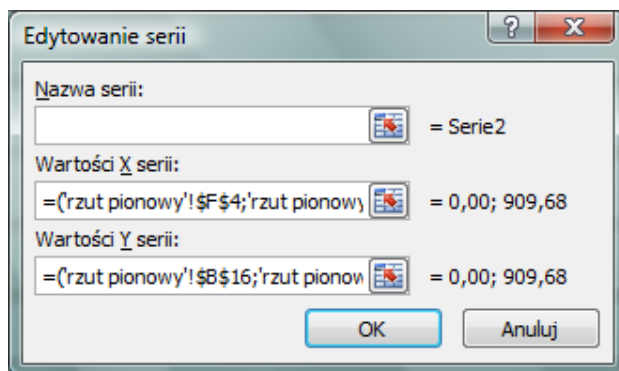
Rysunek 30. Definiowanie pierwszej serii danych

36. W oknie *Wybieranie źródła danych* jest zdefiniowana pierwsza seria, czyli współrzędne wszystkich położenia x,y (czyli tor) ciała w rzucie pionowym przy zadanych parametrach. Dodamy jeszcze drugą serię danych, która będzie się składała z zaledwie dwóch punktów. Będzie ona – przede wszystkim – pokazywała bieżące położenie ciała. Drugim jej zadaniem będzie wyskalowanie wykresu w taki sposób, by na obu osiach układu współrzędnych był dokładnie taki sam zakres wartości. W oknie *Wybieranie źródła danych* kliknij na przycisku *Dodaj* (Rysunek 31).



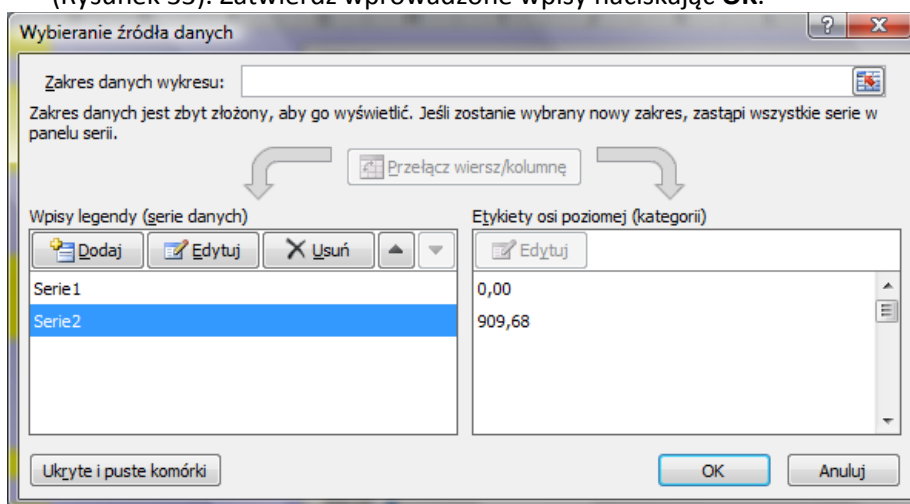
Rysunek 31. Okno *Wybieranie źródła danych* ze zdefiniowaną pierwszą serią

37. W oknie *Edytowanie serii* (Rysunek 32) pole *Nazwa serii* pozostaw puste. Ustaw kursor w polu *Wartości X serii* i zaznacz komórkę F4 oraz dodatkowo – przytrzymując wciśnięty klawisz **Ctrl** – komórkę F5. Następnie przestaw kursor do pola *Wartości Y serii* i zaznacz komórkę B16 oraz – dodatkowo, z wciśniętym klawiszem **Ctrl** – komórkę F5. Potwierdź ustawienia w oknie *Edytowanie serii* naciskając **OK**.



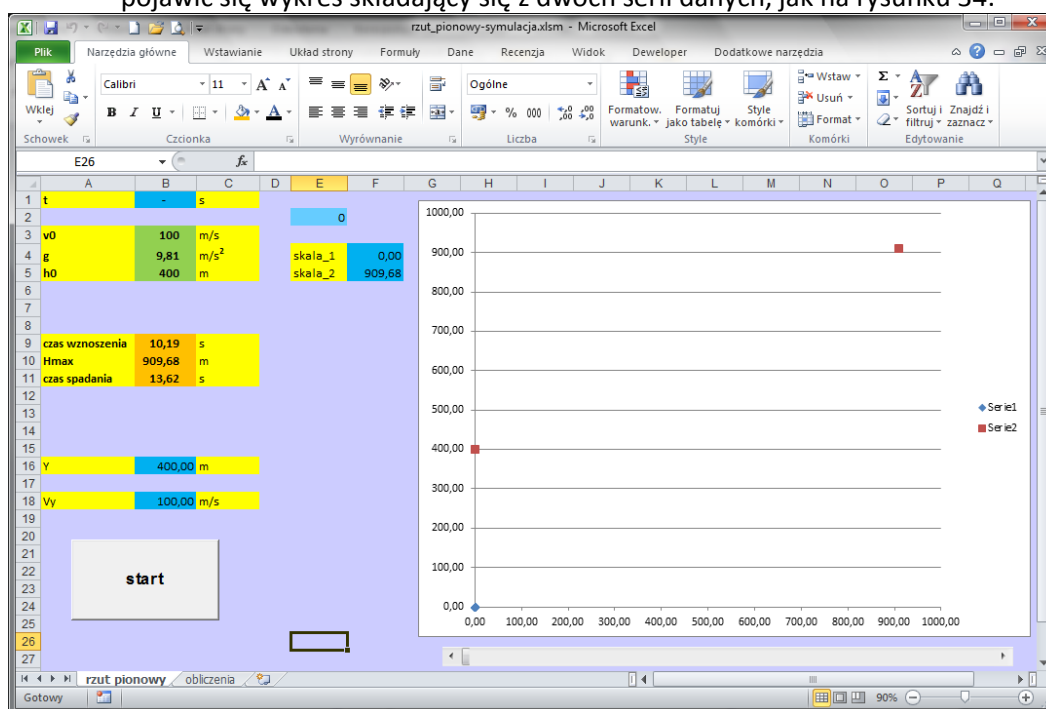
Rysunek 32. Definiowanie drugiej serii danych

38. Spójrz na okno *Wybieranie źródła danych*. Pojawiły się w nim wpisy obu serii danych (Rysunek 33). Zatwierdź wprowadzone wpisy naciskając **OK**.



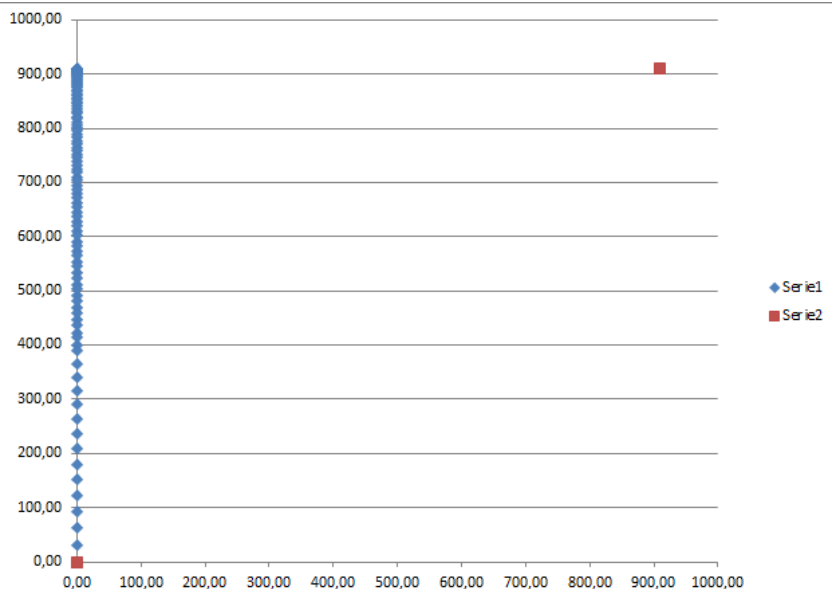
Rysunek 33. Wpisy dwóch serii danych w oknie *Wybieranie serii danych*

39. Po zatwierdzeniu wpisów w oknie *Wybieranie serii danych* w arkuszu powinien pojawić się wykres składający się z dwóch serii danych, jak na rysunku 34.



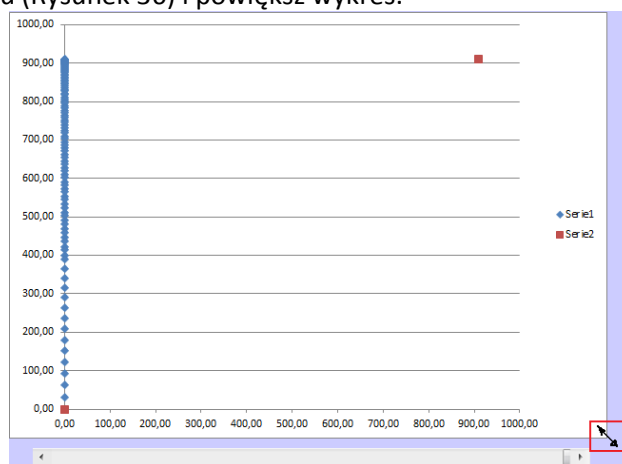
Rysunek 34. Wygląd arkusza *rzut pionowy* z wykresem

40. Wygląd wykresu w dużym stopniu będzie zależał od tego, co znajduje się w arkuszu *obliczenia* w komórkach A1:B101. Jeśli są w nich zera, na wykresie będą zaledwie 3 punkty. Jeśli uruchomisz teraz przyciskiem **start** symulację rzutu pionowego, do komórek A1:B101 w arkuszu *obliczenia* zostaną podstawione niezerowe wartości i wykres będzie wyglądał jak na rysunku 35. Jeśli wprowadzisz zmianę którejkolwiek wartości w komórkach z zielonym tłem, komórki w arkuszu *obliczenia* zostaną wyzerowane (za sprawą makra *Worksheet_change*) i wykres ponownie będzie wyglądał jak na rysunku 34, dopóki ponownie nie uruchomisz symulacji przyciskiem **start** itd.



Rysunek 35. Wykres przedstawiający symulację rzutu pionowego

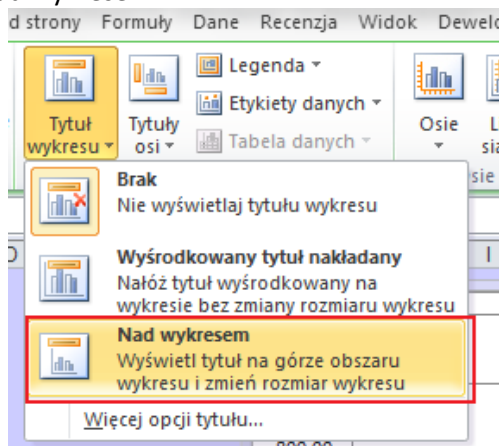
41. Konieczne będzie sformatowanie wykresu, bowiem jego obecny wygląd znacznie odbiega od wzorca, do którego dążymy. Pierwsze, co możemy zrobić, to powiększenie obszaru wykresu tak, aby znalazł się w zakresie wierszy 1:25 i kolumn G:R. Operując kursorem myszki w rogach wykresu uzyskaj znacznik do zmiany rozmiaru (Rysunek 36) i powiększ wykres.



Rysunek 36. Najedź na róg wykresu – pojawi się znacznik służący do zmiany jego rozmiaru

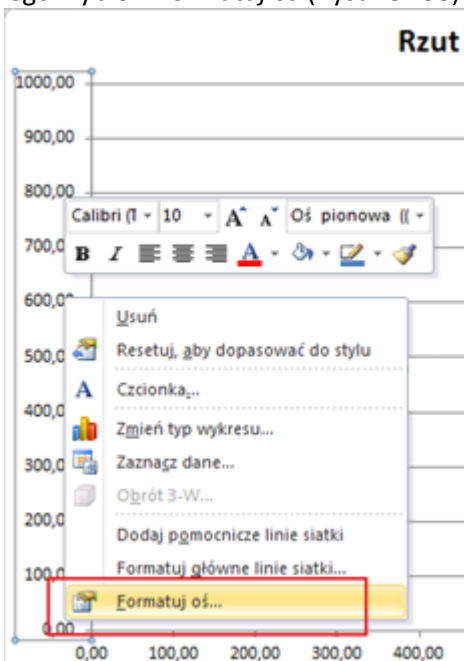
42. Przejdź na kartę *Układ* (znajdującą się w grupie *Narzędzia wykresów* na końcu wstążki; grupa narzędzi wykresów będzie widoczna pod warunkiem, że będzie aktywny wykres) i w sekcji *Etykiety* wybierz ikonę *Legenda*, a następnie, z wewnętrznej listy, *Brak*. W ten sposób wyłączymy wyświetlanie legendy na wykresie. Nie jest nam ona potrzebna, bowiem nie niesie żadnej istotnej informacji.

43. Na karcie *Układ* w sekcji *Etykiety* wybierz *Tytuł wykresu*, a następnie *Nad wykresem*, jak na rysunku 37. Wpisz tekst „Rzut pionowy” i naciśnij **Enter**. Tytuł powinien pojawić się nad wykresem.



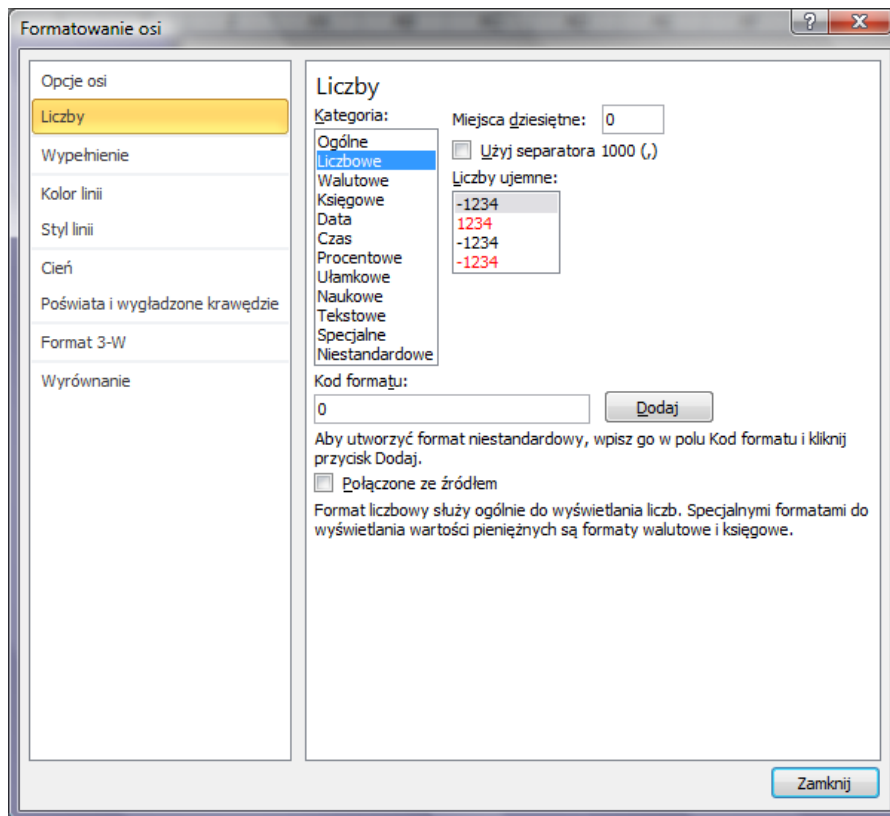
Rysunek 37. Dodawanie tytułu wykresu

44. Kliknij na jednej z wartości osi pionowej (powinna zaznaczyć się cała oś). Przejdź na kartę *Narzędzia główne* i w grupie *Czcionka* włącz pogrubienie (ikona z literą **B**). Możesz również zmienić rozmiar czcionki na większy (np. 12), żeby opisy osi były bardziej czytelne.
45. Kliknij jeszcze raz na jednej z wartości, ale tym razem prawym klawiszem myszki, i z menu podręcznego wybierz *Formatuj oś* (Rysunek 38).



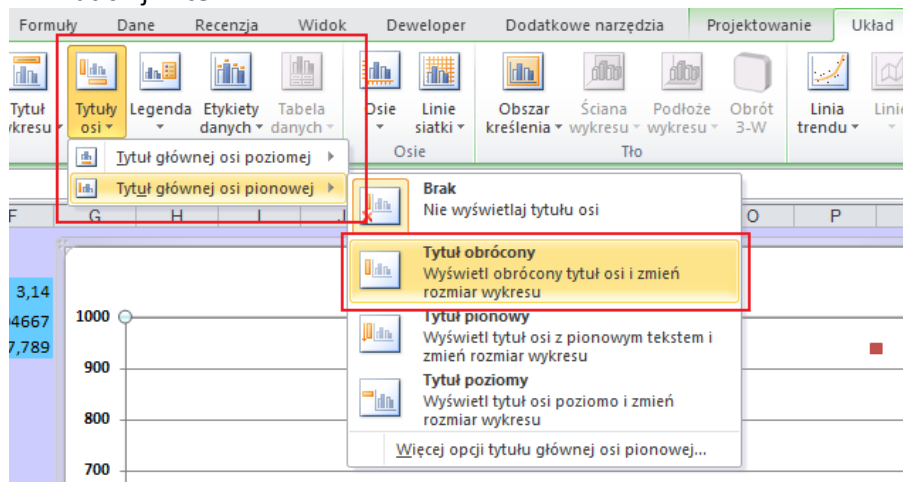
Rysunek 38. Menu podręczne osi pionowej

46. Po lewej stronie okna *Formatowanie* osi wybierz kartę *Liczby*. Na karcie *Liczby*, w prawej części okna, wybierz kategorię *Liczbowe*, w polu *Miejsca dziesiętne* wpisz 0, wyłącz separator tysięcy (Rysunek 39). Na koniec kliknij na przycisku **Zamknij**. Podobnie postępując ustaw identyczne opcje dla osi poziomej.



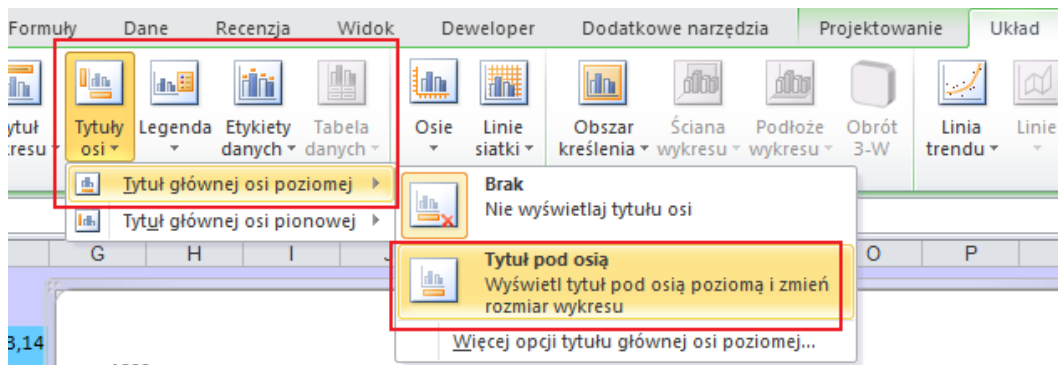
Rysunek 39. Opcje wyświetlania liczb na osi pionowej

47. Przejdź na kartę *Układ* i w sekcji *Etykiety* wybierz *Tytuły osi*, a następnie *Tytuł głównej osi pionowej* i z wewnętrznej listy *Tytuł obrócony* (Rysunek 40). Wpisz tekst „Y[m]” i naciśnij **Enter**.



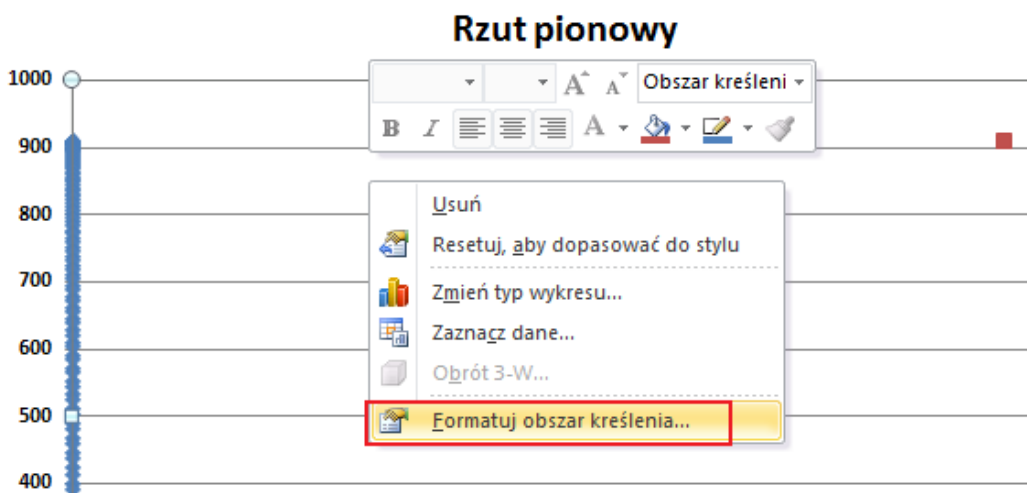
Rysunek 40. Dodanie tytułu osi pionowej

48. Ponownie wybierz polecenie *Tytuły osi*, a następnie *Tytuł głównej osi poziomej* i – z wewnętrznej listy – *Tytuł pod osią* (Rysunek 41). Wpisz tekst „X[m]” i naciśnij **Enter**.



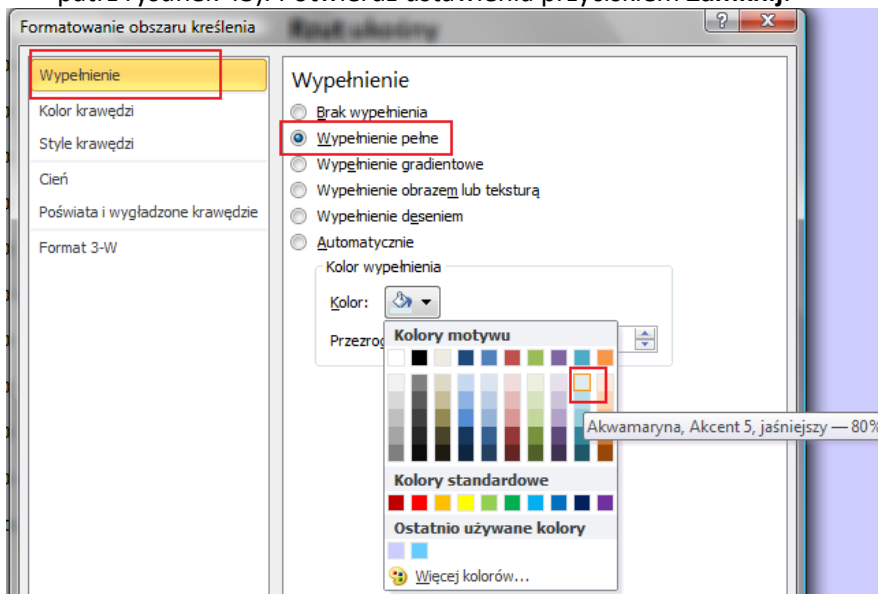
Rysunek 41. Dodanie tytułu osi poziomej

49. Kliknij prawym klawiszem myszki gdzieś w tle wykresu i z menu podręcznego wybierz *Formatuj obszar kreślenia* (Rysunek 42).



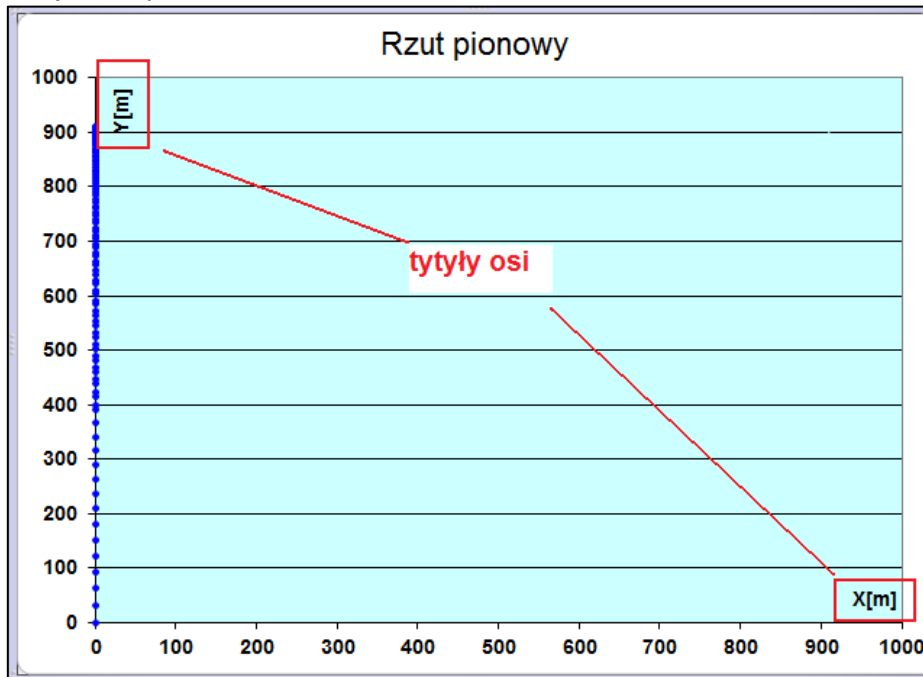
Rysunek 42. Wybór polecenia do sformatowania tła wykresu

50. W oknie *Formatowanie obszaru kreślenia*, na karcie *Wypełnienie* wybierz opcję *Wypełnienie pełne*, a następnie jakiś jasnoniebieski kolor (np. Akwamaryna, Akcent 5 – patrz rysunek 43). Potwierdź ustawienia przyciskiem **Zamknij**.



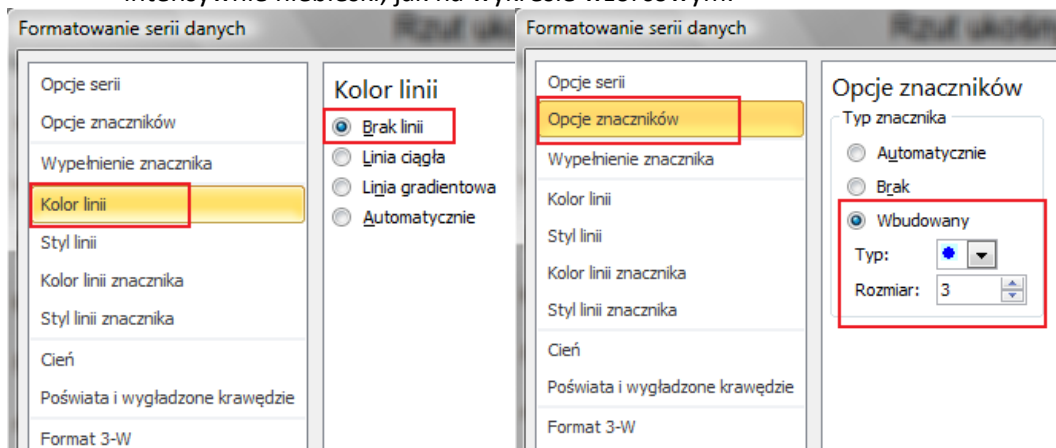
Rysunek 43. Wybór koloru

51. Metodą „przeciągnij i upuść” (chodzi o „złapanie” kursorem myszki jakiegoś obiektu i przeciągnięcie go w odpowiednie miejsce) ustaw podpisy osi, aby całość wyglądała jak na rysunku 44.



Rysunek 44. Wykres po sformatowaniu osi oraz zmianie koloru tła

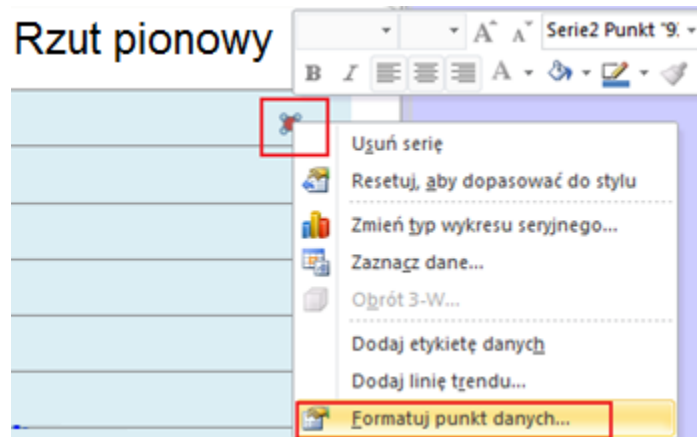
52. Kliknij prawym klawiszem myszki na jednym ze znaczników pierwszej serii danych (punkty z arkusza *obliczenia*) i z menu podręcznego wybierz polecenie *Formatuj serię danych*.
53. W oknie *Formatuj serię danych* wprowadź ustawienia jak na rysunku 45: karta *Kolor linii* wybierz: brak linii, karta *Opcje znaczników*: typ wbudowany, okrągły, o rozmiarze 3. Na karcie *Wypełnienie znacznika* wybierz odpowiedni kolor – na przykład intensywnie niebieski, jak na wykresie wzorcowym.



Rysunek 45. Formatowanie znaczników pierwszej serii danych

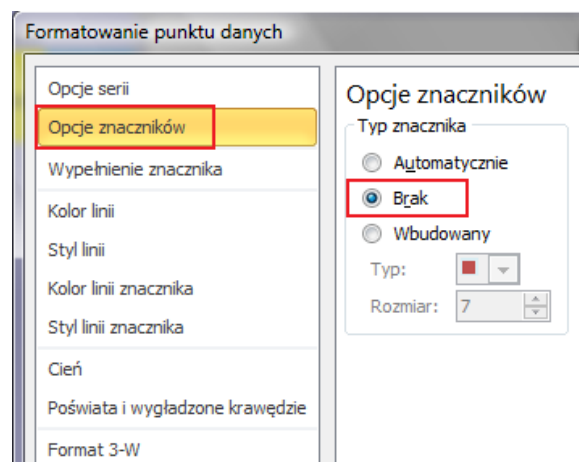
54. „Ukryjemy” teraz punkt drugiej serii danych, który nie przedstawia żadnych istotnych treści fizycznych, a służy nam jedynie do skalowania wykresu. W tym celu kliknij prawym klawiszem na punkcie w prawym górnym rogu drugiej serii danych, po chwili kliknij ponownie i z menu podręcznego wybierz polecenie *Formatuj punkt danych* (Rysunek 46).

Uwaga: jeśli klikniesz raz na jednym ze znaczników serii danych – zaznaczy się cała seria. Jeśli klikniesz ponownie (ale nie za szybko!), zaznaczy się jeden wybrany punkt.



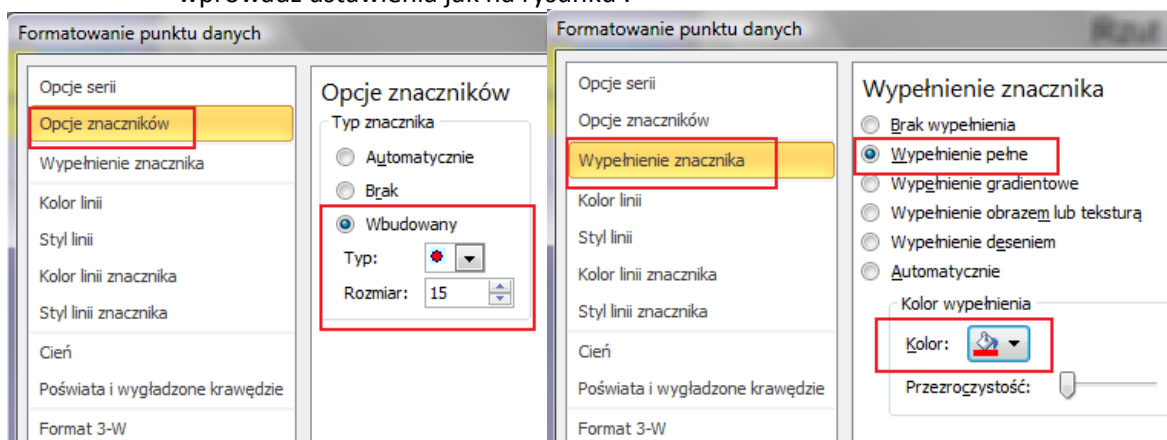
Rysunek 46. Menu podręczne punktu drugiej serii danych

55. W oknie *Formatuj punkt danych* na karcie *Opcje znaczników* wybierz: *Brak* (Rysunek 47).



Rysunek 47. Formatowanie punktu drugiej serii danych

56. Został nam jeszcze jeden punkt do sformatowania – ten, który symuluje ruch ciała w rzucie pionowym. Zaznacz ten punkt (przypominamy, że musisz kliknąć na nim wolno dwa razy, jeśli klikniesz raz – zaznaczą się wszystkie punkty należące do tej serii). Kliknij na zaznaczonym punkcie prawym przyciskiem myszki i z menu podręcznego wybierz *Formatuj punkt danych*. W oknie *Formatuj punkt danych* wprowadź ustawienia jak na rysunku .



Rysunek 48. Opcje punktu drugiej serii symulującego ruch ciała

----- KONIEC -----

Dodatek – podstawowe informacje o VBA

Makropolecenia (makra) składają się z serii poleceń języka programowania zintegrowanego z Excelem. Językiem tym jest **Visual Basic for Application**, skrótowo nazywany **VBA**. Jest to obiektowy język programowania umożliwiający zarówno korzystanie z właściwości arkusza, jak również wywoływanie funkcji systemowych Windows.

Makro jest programem, który wykonuje określone zadania. W Excelu spotykamy się z dwoma rodzajami aplikacji użytkownika:

- ❑ procedury, których użycie pozwala na zautomatyzowanie jednej lub wielu następujących po sobie czynności. Rozpoczynają się od słowa kluczowego *Sub*, po którym następuje nazwa procedury nadana przez użytkownika, a kończą słowami kluczowymi *End Sub*.
- ❑ funkcje, czyli makra wykonujące obliczenia w komórce arkusza. Wynik działania funkcji użytkownika – podobnie jak w przypadku korzystania z funkcji wbudowanych – jest umieszczany w komórce arkusza, w której funkcja została wywołana. Łatwo je odróżnić od procedur typu *Sub*, gdyż rozpoczynają się od słowa kluczowego *Function*, po którym następuje nazwa funkcji nadana przez użytkownika, a kończą słowami kluczowymi *End Function* – nie będziemy z nich korzystać w ćwiczeniu – symulacji rzutu pionowego.

Edytor VBA wywołujemy naciskając kombinację klawiszy **Alt+F11**. Jest on wyposażony w polecenia i paski narzędzi ułatwiające tworzenie, analizę i uruchamianie makr. Gotowe makra można również uruchamiać na przykład: poleceniem ze wstążki **WIDOK→MAKRA→WYŚWIETL MAKRA**, klikając myszą na specjalnie utworzonym przycisku lub naciskając ustaloną kombinację klawiszy (tzw. skrót klawiaturowy).

Kod źródłowy makra ma postać tekstu, który można również edytować zwyczajnym edytorem. Aby podejrzeć tekst źródłowy makropoleceń zapisanych w arkuszu należy otworzyć to makro w edytorze VBA. W tym celu można kliknąć polecenie **WIDOK→MAKRA→WYŚWIETL MAKRA**, wybrać z listy odpowiednie makropolecenie i nacisnąć przycisk **Edycja**. Można również otworzyć edytor VBA kombinacją klawiszy **Alt+F11**, odnaleźć odpowiednie makro (może one być przypisane do konkretnego arkusza, całego skoroszytu, lub znajdować się w specjalnym „kontenerze” zwanym *Module*, który jest przypisany do całego skoroszytu. Lokalizacja makra w dużym stopniu zależy od tego, co ma ono za zadanie wykonać. Do konkretnych arkuszy są często przypisywane tzw. procedury zdarzeniowe. Bardziej ogólnie działające procedury, często operujące na danych z wielu arkuszy, wstawiane są właśnie w dodatkowych „modułach”. Podobnie jest z funkcjami tworzonymi przez użytkownika. Informacja w edytorze VBA, w przeciwieństwie do normalnego arkusza Excela, nie jest podzielona na komórki, lecz przedstawiona w formie zwartej ciągu tekstu.

Typy danych

Zadaniem procedur VBA jest manipulowanie różnego rodzaju danymi. Typ danych określa, w jaki sposób są one przechowywane w pamięci komputera. Informuje też użytkownika, z jakiego typu danymi ma do czynienia. Dla przykładu, dane mogą być przechowywane jako liczby (całkowite lub rzeczywiste), łańcuchy znaków (teksty), wartości logiczne (*True* (Prawda) lub *False* (Fałsz)), daty itd.

Jeśli, pisząc procedury, nie określimy typu danych, to Visual Basic automatycznie zastosuje typ *Variant*. *Variant* sam sprawdza, z jaką wartością ma do czynienia i „dopasowuje się” do niej. Może on zastępować dowolny typ danych, ale używanie go powoduje marnowanie czasu i pamięci komputera, szczególnie w przypadku tworzenia długich i skomplikowanych procedur. Przypisanie określonych typów danych do zmiennych używanych przez nas w kodzie przyspiesza wykonywanie programu i oszczędza pamięć, ponieważ Excel rezerwuje tylko tyle pamięci, ile jest potrzebne. Poniżej przedstawiamy zestawienie przykładowych typów danych (podamy je informacyjnie):

- ❑ *String* łańcuch tekstowy o długości od 0 do 65 535 znaków;

- ❑ *Byte* liczba całkowita z przedziału od 0 do 255
- ❑ *Integer* liczba całkowita z przedziału od -32 768 do 32 767;
- ❑ *Long* liczba całkowita długa z zakresu od -2 147 483 648 do 2 147 483 648;
- ❑ *Currency* liczba całkowita z czterema miejscami po przecinku od -922.337.203.685.477,5808 do 922.337.203.685.477,5807;
- ❑ *Date* data w przedziale od 1/1/100 do 31/12/9999;
- ❑ *Single* liczba rzeczywista (może mieć część ułamkową) od -3,402823E38 do 1,401298E45;
- ❑ *Double* liczba rzeczywista dwukrotnie większa od liczby *Single* z zakresu -1,79769313486232E308 do -4,94065645841247E-324 dla liczb ujemnych i od 4,94065645841247E-324 do 1,79769313486232E308 dla liczb dodatnich;
- ❑ *Variant* dane dowolnego typu;
- ❑ i wiele innych typów, również definiowanych przez użytkownika.

Zmienne i ich deklaracje

Deklarowanie zmiennych jest dobrym zwyczajem, ponieważ ułatwia analizę kodu programu, pomaga też łatwiej wychwycić niektóre błędy. Zmienna jest wyrazem, skrótem, który reprezentuje wynik działania jakiejś instrukcji. Jak sama nazwa wskazuje, wartość zmiennej może ulegać zmianie w czasie wykonywania programu.

Nazwy nadawane zmiennym nie mogą być zupełnie dowolne. Podlegają one pewnym ścisłym regułom. Mogą składać się z liter, cyfr i niektórych znaków przestankowych z wyjątkiem znaków: . # \$ % & @ ! Nazwa zmiennej nie może się zaczynać od cyfry i nie może zawierać spacji! Visual Basic zawiera też listę słów w języku angielskim, które nie mogą być używane jako nazwa zmiennej. Są to tzw. słowa kluczowe języka, takie jak np. *Abs*, *Date*, *False*, *Select*, *Static*, i wiele innych. Visual Basic nie rozróżnia małych i dużych liter w nazwach zmiennych.

Jak już wcześniej wspomnieliśmy, każda zmienna posiada określony typ. Deklarowanie nazw zmiennych i ich typów przed pierwszym użyciem w procedurze daje dwie korzyści:

- ❑ przyspiesza wykonanie procedury i oszczędza pamięć,
- ❑ zapobiega błędom wynikającym z nieprawidłowego wpisania nazwy zmiennej.

Deklaracja zmiennej polega na wpisaniu instrukcji *Dim* (skrót od Dimension), po której następuje nazwa zmiennej oraz jej typ. Przyjęta składnia deklaracji zmiennych jest następująca:

Dim zmienna *As* typ

gdzie **zmienna** stanowi nazwę zmiennej, a **typ** określa typ danych, które będą przechowywane w zmiennej. Na przykład polecenie:

Dim Nazwisko *As* String

deklaruje zmienną *Nazwisko* jako zmienną typu *String* (tekst, łańcuch znaków). Natomiast deklaracja:

Dim Nazwisko

deklaruje zmienną *Nazwisko* jako zmienną typu *Variant*.

Instrukcja warunkowa *If ... Then*

If ... Then jest najprostszą strukturą do podejmowania decyzji w Visual Basicu. Umożliwia ona wykonanie jednej lub więcej instrukcji w zależności od podanego warunku. **Warunek** może być liczbą lub tekstem, który przekazuje wartość logiczną *True* lub *False*. **Instrukcje** – jedna lub więcej dowolnych instrukcji Visual Basicu, które procedura ma wykonać kiedy **warunek** okaże się prawdą. Struktura *If Then* może być zapisana następujący sposób:

```
If warunek Then
    [instrukcje]
End If
```

Słowa kluczowe *If* i *End if* są tu jakby klamrą spinającą cały blok instrukcji. Dokładnie wiadomo kiedy blok się zaczyna i gdzie się kończy.

W ogólności zdarza się, że potrzebne są jedne **instrukcje**, jeśli **warunek** jest prawdziwy oraz inne, jeżeli **warunek** jest fałszywy. Wówczas pomocna będzie struktura określana jako *If ... Then ... Else* (podajemy ją tylko informacyjnie, nie stosujemy w niniejszym opracowaniu):

```
If warunek Then
    [instrukcje, gdy warunek jest prawdą]
Else
    [instrukcje, gdy warunek jest fałszywy]
End If
```

Powyższa struktura blokowa wykonuje **instrukcję** lub blok instrukcji, gdy **warunek** jest prawdziwy i jedną **instrukcję** lub blok instrukcji, gdy **warunek** jest fałszywy. Jeżeli warunek jest prawdziwy, to Visual Basic wykona instrukcje zawarte pomiędzy *Then* i *Else*, natomiast opuści instrukcje znajdujące się pomiędzy *Else* i *End If*, a potem wykona dalsze instrukcje procedury. Jeżeli warunek jest fałszywy, to Visual Basic opuści instrukcje znajdujące się pomiędzy *Then* i *Else*, wykona instrukcje zawarte między *Else* i *End If*, a następnie wykona dalsze instrukcje procedury.

Przykład zastosowania znajduje się na końcu.

Pętla liczona *For ... Next*

Pętla *For ... Next* wykona dany blok **instrukcji** z góry określoną ilość razy. Oto jak wygląda składnia pętli:

```
For licznik = start To stop (Step krok)
    [instrukcje]
Next licznik
```

Licznik jest dowolną nazwą zmiennej używanej do przechowywania liczby powtórzeń. **Start** określa początkową wartość licznika, **stop** – maksymalną liczbę powtórzeń. Słowo kluczowe *Next* odsyła ponownie do pierwszej linijki pętli i sprawdzenia wartości licznika. Jeżeli liczba w liczniku okaże się większa niż wartość maksymalna (*stop*), to nastąpi wyjście z pętli. **Krok** określa o ile ma się zmienić wartość licznika przy przejściu do następnego powtórzenia pętli. Użycie wyrażenia podanego w nawiasie, tzn. *Step krok*, jest opcjonalne. Jeśli go pominiemy, za każdym razem, gdy Visual Basic wykona instrukcje w pętli, wartość licznika będzie zwiększana o jeden.

Przykład zastosowania pętli znajduje się na końcu.

Model obiektowy. Odwołania do obiektów

Excel posiada ponad 100 **obiektów**. Przykładowe obiekty to:

- ❑ skoroszyt (*Workbook*),
- ❑ arkusz (*Worksheet*),

- ❑ zakres komórek w arkuszu (*Range*),
- ❑ komórka (*Cells*), alternatywa dla obiektu *Range*, w niektórych przypadkach łatwiejsza w użyciu,

i wiele, wiele innych.

Obiekty są zorganizowane w pewną hierarchię. Niektóre obiekty zawierają w sobie inne obiekty. Excel jest obiektem zwanym aplikacją (*Application*) i jest on obiektem nadrzędnym w stosunku do innych. Obiekt *Application* zawiera w sobie obiekty, takie jak skoroszyty (*Workbooks*). Skoroszyt zaś zawiera inne obiekty, takie jak arkusze (*Worksheets*) czy wykresy (*Charts*).

Obiekt *Range* może oznaczać pojedynczą komórkę lub zakres komórek arkusza. Jak wyżej wspomnieliśmy, znajduje się on wewnątrz większego obiektu, jakim jest *Worksheet*.

Range pozwala na odwołanie do komórki arkusza lub większego zakresu komórek za pomocą adresu (np. A1, B5, A2:A10, itd.). Jeżeli chcemy odwołać się do komórki B5 pierwszego arkusza skoroszytu o nazwie *ćwiczenia* (za pomocą obiektu *Range*), żeby, na przykład, ją zaznaczyć, musimy w procedurze VBA wpisać następującą instrukcję:

```
Workbooks("ćwiczenia").Worksheets(1).Range("B5").Select
```

Jak widzisz – poszczególne elementy w hierarchii są rozdzielone kropką. Aby wybrać zakres komórek (np. od C6 do C10) wpisz:

```
Range("C6:C10").Select
```

Właściwość *Cells* wymaga zazwyczaj podania dwóch argumentów. Argument pierwszy wskazuje numer wiersza, natomiast drugi, to numer kolumny. W tej konwencji odwołanie do komórki B5, która znajduje się na przecięciu drugiej kolumny i piątego wiersza, będzie wyglądało następująco:

```
ThisWorkbook.Worksheets(1).Cells(5,2)
```

W powyższej instrukcji obiektem nadrzędnym jest *ThisWorkbook*. Oznacza dosłownie „ten skoroszyt”, ten – w którym piszemy makro. Taki sposób odwołania do skoroszytu jest alternatywą dla zapisu *Workbooks("nazwa")*. W przypadku, kiedy makro operuje na wielu skoroszytach, łatwiej jest odróżnić w kodzie programu te instrukcje, które mają się odwoływać do skoroszytu, w którym jest makro.

PRZYKŁADY:

Przykład zastosowania instrukcji warunkowej *If*: założmy, że w komórce A1 jest jakaś liczba, którą chcemy podzielić przez jakąś inną liczbę znajdującą się w komórce A2. Wynik dzielenia chcemy uzyskać w komórce B1.

Zwróć uwagę, że gdyby w komórce A2 użytkownik wpisał zero, wówczas w wyniku wykonania dzielenia w komórce wynikowej – B1 – pojawi się błąd!

Chcemy aby nasza procedura działała następująco: jeżeli liczba w komórce A2 jest różna od zera, wówczas wykonaj dzielenie i wynik wpisz do B1. W przeciwnym wypadku (oznacza to, że w komórce A2 jest wpisane zero), w komórce B1 wyświetli się komunikat w postaci tekstu „Błąd! Nie dziel przez zero!”.

```
If Range(„A2”) <> 0 Then  
    Range(„B1”) = Range(„A1”) / Range(„A2”)  
Else  
    Range(„B1”) = „Błąd! Nie dziel przez zero!”  
End If
```

Brak w powyższym przykładzie odwołania do nadrzędnych obiektów w stosunku do *Range* w postaci arkusza czy skoroszytu (na przykład: *ThisWorkbook.Worksheets(1).Range(„B1”)*) oznacza, że procedura ma się wykonać w aktywnym skoroszytu, w aktywnym arkuszu.

Podamy jeszcze, jak należałoby zapisać tę samą instrukcję używając obiektu *Cells*:

```

If Cells(2,1) <> 0 Then
    Cells(1,2) = Cells(1,1) / Cells(2,1)
Else
    Cells(1,2) = "Błąd! Nie dziel przez zero!"
End If

```

Przykład zastosowania pętli For Next: chcemy wypełnić co drugą komórkę z zakresu od A2 do A10 wartościami z których każda jest 100 razy większa niż numer wiersza komórki, do której trafia. Wartość **stop** będzie równa 10 – jest to maksymalna wartość **licznika**. **Licznik** to zmienna reprezentująca obroty pętli – najczęściej nazywamy ją „i”. **Step**, czyli krok, informuje nas o ile różni się każda następną wartość licznika od poprzedniej. **Start** to wartość początkowa **licznika**, u nas wynosi 2.

```

For i = 2 To 10 Step 2
    Cells(i, 1) = i * 100
Next i

```

Po wejściu do pętli zmienna *i* przyjmuje wartość 2. Jeżeli jest ona mniejsza (lub równa) 10 – więc pętla się wykonuje: do komórki o numerze kolumny „1” (czyli A) i numerze wiersza równym bieżącej wartości *i*, czyli 2, podstawia się wynik działania $i*100$, czyli przy pierwszym obrocie naszej pętli $2*100=200$.

Licznik zwiększa się – *i* staje się większe o 2.

W drugim obrocie pętli *i* równa się więc 4. Następuje sprawdzenie czy 4 jest mniejsze lub równe 10, jeśli tak, do kolejnej komórki (o numerze kolumny 1 i numerze wiersza 4 – takim jak *i*) wstawia wartość $i*100=4*100=400$.

Itd. aż bieżąca wartość **licznika** przekroczy wartość **stop**. Wtedy pętla kończy działanie.

Wynik ćwiczenia:

	A	B	C	D
1				
2	200			
3				
4	400			
5				
6	600			
7				
8	800			
9				
10	1000			
11				